UNCLASSIFIED

AD NUMBER
AD917002
LIMITATION CHANGES
TO: **Approved for public release; distribution is unlimited.**
FROM: **Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; NOV 1973. Other requests shall be referred to Air Force Armament Laboratory, Eglin AFB, FL 32542.**
AUTHORITY
AFATL ltr, 14 Nov 1975

THIS PAGE IS UNCLASSIFIED

AFATL-TR-73-221

A DIGITAL COMPUTER PROGRAM FOR EXTRACTING AERODYNAMIC COEFFICIENTS FROM SIX-DEGREE-OF-FREEDOM DYNAMIC DATA

UNIVERSITY OF FLORIDA

TECHNICAL REPORT AFATL-TR-73-221

NOVEMBER 1973

Distribution limited to U. S. Government agencies only; this report documents test and evaluation; distribution limitation applied November 1973. Other requests for this document must be referred to the Air Force Armament Laboratory (DLMA), Eglin Air Force Base, Florida 32542.

AIR FORCE ARMAMENT LABORATORY

AIR FORCE SYSTEMS COMMAND • UNITED STATES AIR FORCE

EGLIN AIR FORCE BASE, FLORIDA

A Digital Computer Program For Extracting Aerodynamic Coefficients From Six-Degree-Of-Freedom Dynamic Data

F. W. Steinbauer
M. H. Clarkson
T. E. Bullock

Distribution limited to U. S. Government agencies only; this report documents test and evaluation; distribution limitation applied November 1973. Other requests for this document must be referred to the Air Force Armament Laboratory (DLMA), Eglin Air Force Base, Florida 32542.

FOREWORD

This analysis was conducted by the University of Florida, Gainesville, Florida, under Contract F08635-73-C-0009, with the Air Force Armament Laboratory, Eglin Air Force Base, Florida. The effort was conducted during the period March 1972 to March 1973. Dr. George B. Findley (DLMA) was program manager for the Armament Laboratory. This work was partially supported by the Air Force Office of Scientific Research (AFOSR) under its project 9871.

The principal investigators for the University of Florida were Drs. T. E. Bullock and M. H. Clarkson.

This technical report has been reviewed and is approved.

Richard M Keller

RICHARD M. KELLER, Colonel, USAF
Chief, Air-to-Surface Modular
Guided Weapons Division

ABSTRACT

The development of a digital computer program to extract aerodynamic coefficients from dynamic data for six-degree-of-freedom systems is presented. The derivation of a system mathematical model is discussed in detail. Results and associated problems of extracting coefficients from one-, two-, three-,and six-degree-of-freedom systems data are also presented.

Distribution limited to U. S. Government agencies only; this report documents test and evaluation; distribution limitation applied November 1973. Other requests for this document must be referred to the Air Force Armament Laboratory (DLMA), Eglin Air Force Base, Florida 32542.

(The reverse of this page is blank)

TABLE OF CONTENTS

(The reverse of this page is blank)

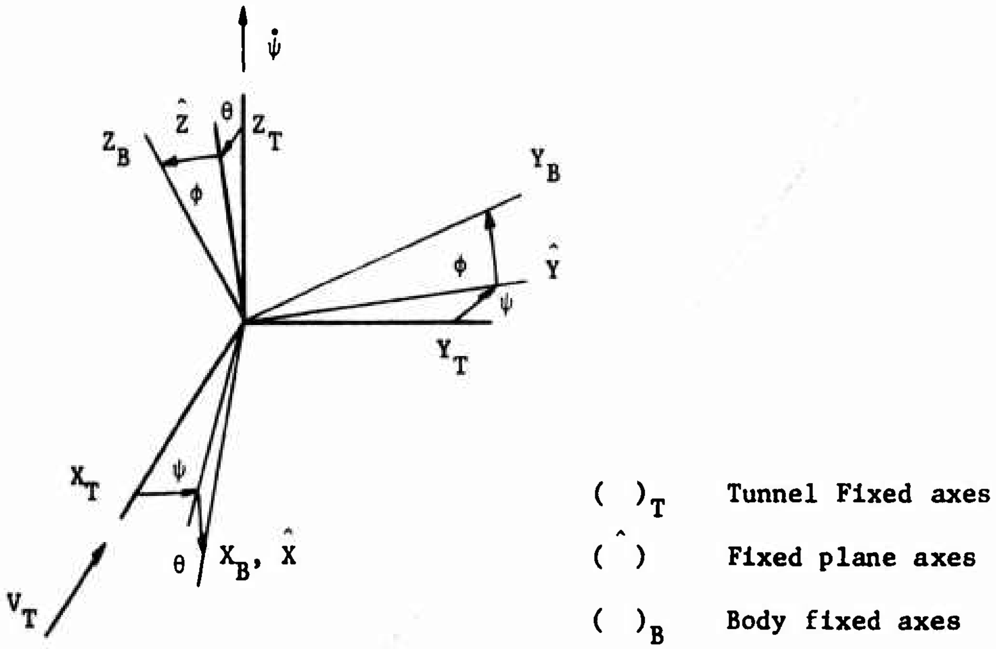
SECTION I

INTRODUCTION

Early methods of extracting aerodynamic coefficients from dynamic data required assumptions and limitations so that the equations of motion could be solved in closed form. Therefore, usually only linear aerodynamics were considered. As a result, the range of motions and the number of coefficients extracted were severely limited.

The method of extracting coefficients by means of parametric differentiation developed by Chapman and Kirk(1) is not restricted by the requirement of linear aerodynamics. In this report the method of parametric differentiation is used to develop a six-degree-of-freedom digital computer program to extract aerodynamic coefficients from free flight data. The program is an extension of the one- and three-degree-of-freedom programs of Daniel and Bullock(2), respectively, and draws on their experience in developing those programs.

The equations of motion for the six-degree-of-freedom mathematical model were developed so that the aerodynamic coefficients presented by Holmes(3) may be used. The model, which is intended for use in an aeroballistic wind tunnel range, uses a fixed plane axis system to represent the angular orientation of the body with respect to a tunnel fixed axis system. By definition, the fixed plane axes are free to pitch and yaw with the body but do not roll with the body. The relationship between the axis systems is depicted in the diagram below.



SECTION II

EQUATIONS OF MOTION

The derivation of the equations of motion for the six-degree-of-freedom model used in the program assumes that the missile is regarded as a rigid axisymmetric body moving with velocity V_T relative to a wind tunnel axis system. In addition, the body fixed axes are chosen to coincide with the principal axes of the missile.

The equations for translational and angular motion, based on Newton's second law, may be written as

$$m \frac{d}{dt} \bar{V}_T = \bar{F}_T \tag{1}$$

and

$$\frac{d\bar{h}}{dt} + \bar{\omega}_{FP} \times \bar{h} = \bar{M}_{FP} \tag{2}$$

where: $\bar{F}_T$ is the resultant external force.

$\bar{\omega}_{FP}$ is the angular velocity of the fixed plane axes with respect to the tunnel fixed axes.

$\bar{h}$ is the moment of momentum.

$\bar{M}_{FP}$ is the resultant external moment.

The equations of motion above provide a form suitable for fitting to the data. First, however, it is necessary to define the orientation of the fixed plane axes with respect to the aerodynamic data axes, which contain the cameras that recorded the motion and position of the body during flight, and then to define the tunnel axes (assumed inertial) with respect to the fixed plane axes.

Choosing the body fixed axes to lie along the principal axes of the missile results in the products of inertia being zero. Thus, the angular momentum vector may be expressed in terms of the angular velocity and the moments of inertia.

$$h = h_x \hat{i} + h_y \hat{j} + h_z \hat{k} = I_x \omega_x \hat{i} + I_y \omega_y \hat{j} + I_z \omega_z \hat{k} \tag{3}$$

Recalling the relationship between the fixed plane axes and the body fixed axes, equation (2) may be written in component form

$$
\begin{aligned}
I_x \dot{\omega}_x + \omega_{y_{FP}} I_z \omega_z - \omega_{z_{FP}} I_y \omega_y &= M_{x_{FP}} \\
I_y \dot{\omega}_y + \omega_{z_{FP}} I_x \omega_x - \omega_{x_{FP}} I_z \omega_z &= M_{y_{FP}} \qquad (4) \\
I_z \dot{\omega}_z + \omega_{x_{FP}} I_y \omega_y - \omega_{y_{FP}} I_x \omega_x &= M_{z_{FP}}
\end{aligned}
$$

Recalling that the body fixed axes were principal axes implies that

$$I_y = I_z = I \qquad (5)$$

Now expressing the angular velocity components of the fixed plane axes in terms of the Euler angles yields

$$
\begin{aligned}
\omega_{x_{FP}} &= - \dot{\psi}\ \mathrm{Sin}\ \theta = \hat{p} \\
\omega_{y_{FP}} &= \dot{\theta} = \hat{q} \qquad (6) \\
\omega_{z_{FP}} &= \dot{\psi}\ \mathrm{Cos}\ \theta = \hat{r}
\end{aligned}
$$

and the angular velocity components of the body fixed axes are

$$
\begin{aligned}
\omega_x &= \dot{\phi} - \dot{\psi}\ \mathrm{Sin}\ \theta = p \\
\omega_y &= \dot{\theta} = q \qquad (7) \\
\omega_z &= \dot{\psi}\ \mathrm{Cos}\ \theta = r
\end{aligned}
$$

which have time derivatives

$$
\begin{aligned}
\dot{\omega}_x &= \ddot{\phi} - \ddot{\psi}\ \mathrm{Sin}\ \theta - \dot{\psi}\dot{\theta}\ \mathrm{Cos}\ \theta \\
\dot{\omega}_y &= \ddot{\theta} \qquad (8) \\
\dot{\omega}_z &= \ddot{\psi}\ \mathrm{Cos}\ \theta - \dot{\psi}\dot{\theta}\ \mathrm{Sin}\ \theta
\end{aligned}
$$

Now applying equations (5), (6), (7) and (8) to the first of equation (4) results in

$$I_x\left[\ddot{\phi} - \ddot{\psi}\ \text{Sin}\ \theta - \dot{\psi}\dot{\theta}\ \text{Cos}\ \theta\right] = M_{x_{FP}} \tag{9}$$

Similarly the second of equations (4) becomes

$$I\ddot{\theta} + \dot{\psi}\ \text{Cos}\ \theta\ I_x\left[\dot{\phi} - \dot{\psi}\ \text{Sin}\ \theta\right] + \dot{\psi}\ \text{Sin}\ \theta\ I\,\dot{\psi}\,\text{Cos}\ \theta = M_{y_{FP}} \tag{10}$$

Rearranging terms yields

$$I\ddot{\theta} + \dot{\psi}\ \text{Cos}\ \theta\left[I_x p + I\ \dot{\psi}\ \text{Sin}\ \theta\right] = M_{y_{FP}} \tag{11}$$

or

$$\ddot{\theta} + \left[p\ \frac{I_x}{I} + \dot{\psi}\ \text{Sin}\ \theta\right]\dot{\psi}\ \cos\ \theta = \frac{M_{y_{FP}}}{I} \tag{12}$$

Now operating in the same manner on the third of equations (4) yields

$$I\left[\ddot{\psi}\ \text{Cos}\ \theta - \dot{\psi}\dot{\theta}\ \text{Sin}\ \theta\right] - \dot{\psi}\ \text{Sin}\ \theta\ I\dot{\theta} - \dot{\theta}I_x\left[\dot{\phi} - \dot{\psi}\ \text{Sin}\ \theta\right] = M_{z_{FP}} \tag{13}$$

Rearranging terms yields

$$I\left[\ddot{\psi}\ \text{Cos}\ \theta - \dot{\psi}\dot{\theta}\ \text{Sin}\ \theta\right] - \dot{\theta}\left[p\ I_x + I\dot{\psi}\ \text{Sin}\ \theta\right] = M_{z_{FP}} \tag{14}$$

or

$$\ddot{\psi}\ \text{Cos}\ \theta - \left[p\ \frac{I_x}{I} + 2\dot{\psi}\ \ \text{Sin}\ \theta\right]\dot{\theta} = \frac{M_{z_{FP}}}{I} \tag{15}$$

Consider, now, equation (1) for translational motion. It may be written, directly, in component form as

$$\begin{aligned}\ddot{X} &= \frac{F_{x_T}}{m}\\ \ddot{Y} &= \frac{F_{y_T}}{m}\\ \ddot{Z} &= \frac{F_{z_T}}{m}\end{aligned} \tag{16}$$

The definitions of the resultant aerodynamic forces above and the resultant aerodynamic moments were represented in terms of resultant aerodynamic force and moment coefficients, C_X, C_Y, C_Z and C_L, C_M, C_N which lie along the aerodynamic data axes. For the translational equations of motion it was first necessary to prescribe how the components of each aerodynamic coefficient along the tunnel fixed axes would be determined in terms of the fixed plane axes. Then for all of the equations of motion it was necessary to transform the components along the fixed plane axes in terms of the aerodynamic axes. The transformations for the translational equations of motion were

$$L(\psi) = \begin{bmatrix} \mathrm{Cos}\ \psi & -\ \mathrm{Sin}\ \psi & 0 \\ \mathrm{Sin}\ \psi & \mathrm{Cos}\ \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (17)$$

$$L(\theta) = \begin{bmatrix} \mathrm{Cos}\ \theta & 0 & \mathrm{Sin}\ \theta \\ 0 & 1 & 0 \\ -\ \mathrm{Sin}\ \theta & 0 & \mathrm{Cos}\ \theta \end{bmatrix} \qquad (18)$$

$$L(\hat{\phi}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathrm{Cos}\ \hat{\phi} & \mathrm{Sin}\ \hat{\phi} \\ 0 & -\ \mathrm{Sin}\ \hat{\phi} & \mathrm{Cos}\ \hat{\phi} \end{bmatrix} \qquad (19)$$

The first and second transformations are straightforward transformations through the Euler angles ψ and θ from the tunnel fixed axes to the fixed plane axes. The third transformation is the transformation of the coefficients in terms of the fixed plane axes to the coefficients in terms of the aerodynamic data axes. The application of the transformation matrices yields the equations

$$\begin{bmatrix} \ddot{X}_T \\ \ddot{Y}_T \\ \ddot{Z}_T + g \end{bmatrix} = \frac{QA}{m} \begin{bmatrix} L(\psi)\ L(\theta) \end{bmatrix} \begin{bmatrix} L(\hat{\phi}) \end{bmatrix} \begin{bmatrix} C_X \\ C_Y \\ C_Z \end{bmatrix} \qquad (20)$$

or, in expanded form,

$$\ddot{X}_T = \frac{QA}{m} \left\{ C_X (\mathrm{Cos}\ \theta\ \mathrm{Cos}\ \psi) - C_Y (\mathrm{Sin}\ \psi\ \mathrm{Cos}\ \hat{\phi} + \mathrm{Sin}\ \theta\ \mathrm{Cos}\ \psi\ \mathrm{Sin}\ \hat{\phi}) - C (\mathrm{Sin}\ \psi\ \mathrm{Sin}\ \hat{\phi} - \mathrm{Sin}\ \theta\ \mathrm{Cos}\ \psi\ \mathrm{Cos}\ \hat{\phi}) \right\} \quad (21)$$

$$\ddot{Y}_T = \frac{QA}{m} \left\{ C_X (\mathrm{Cos}\ \theta\ \mathrm{Sin}\ \psi) + C_Y (\mathrm{Cos}\ \psi\ \mathrm{Cos}\ \hat{\phi} - \mathrm{Sin}\ \theta\ \mathrm{Sin}\ \psi\ \mathrm{Sin}\ \hat{\phi}) + C_Z (\mathrm{Cos}\ \psi\ \mathrm{Sin}\ \hat{\phi} + \mathrm{Sin}\ \theta\ \mathrm{Sin}\ \psi\ \mathrm{Cos}\ \hat{\phi}) \right\} \quad (22)$$

$$\ddot{Z}_T = \frac{QA}{m} \left\{ C_X (- \mathrm{Sin}\ \theta) - C_Y (\mathrm{Cos}\ \theta\ \mathrm{Sin}\ \hat{\phi}) + C_Z (\mathrm{Cos}\ \theta\ \mathrm{Cos}\ \hat{\phi}) \right\} - g \quad (23)$$

Before arriving at the final form of the angular equations of motion a transformation to obtain the components of the coefficients along the aerodynamic axes employing the third transformation, equation (19), must be carried out. The resulting equations are

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = QAd\ L(\hat{\phi}) \begin{bmatrix} C_L \\ C_M \\ C_N \end{bmatrix} \quad (24)$$

or, in expanded form,

$$\ddot{\psi} = \left\{ (p \frac{I_x}{I} + 2\ \dot{\psi}\ \mathrm{Sin}\ \theta)\ \dot{\theta} + \frac{QAd}{I} (C_N\ \mathrm{Cos}\ \hat{\phi} - C_M\ \mathrm{Sin}\ \hat{\phi} + C_{M_D} (\frac{\dot{\psi} d}{2V_A})\ \mathrm{Cos}\ \theta \right\} \frac{1}{\mathrm{Cos}\ \theta} \quad (25)$$

$$\ddot{\theta} = -\left(p\frac{I_x}{I} + \dot{\psi}\ \mathrm{Sin}\ \theta\right)\dot{\psi}\ \mathrm{Cos}\ \theta + \frac{QAd}{I}\left(C_M\ \mathrm{Cos}\ \hat{\phi} + C_N\ \mathrm{Sin}\ \hat{\phi} + C_{M_D}\left(\frac{\dot{\theta}d}{2V_A}\right)\right) \qquad (26)$$

$$\ddot{\phi} = \mathrm{Sin}\ \theta\dot{\psi} + \dot{\psi}\dot{\theta}\ \mathrm{Cos}\ \theta + \frac{QAd}{I}(C_L) \qquad (27)$$

The resultant coefficients in the equations of motion are defined in the following fashion:

$$C_X = C_{x_o} + C_{x_{\bar{\alpha}2}}\bar{\alpha}^2$$

$$C_Y = (C_{y_{\bar{\alpha}}}\bar{\alpha} + C_{y_{\bar{\alpha}3}}\bar{\alpha}^3)\ \mathrm{Sin}\ (NF \cdot \phi) + (C_{y_{\bar{\alpha}_p}}\bar{\alpha} + C_{y_{\bar{\alpha}_p 3}}\bar{\alpha}^3)\left(\frac{\dot{\phi}d}{2V_A}\right)$$

$$C_Z = C_{z_{\bar{\alpha}}}\bar{\alpha} + C_{z_{\bar{\alpha}_p 3}}\bar{\alpha}^3$$

$$C_L = (C_{\ell_{\bar{\alpha}}}\bar{\alpha} + C_{\ell_{\bar{\alpha}3}}\bar{\alpha}^3)\ \mathrm{Sin}\ (NF \cdot \phi) + C_{\ell_p}\left(\frac{\dot{\phi}d}{2V_A}\right) \qquad (28)$$

$$C_M = C_{M_{\bar{\alpha}}}\bar{\alpha} + C_{M_{\bar{\alpha}3}}\bar{\alpha}^3$$

$$C_{M_D} = C_{m_{q_o}} + C_{M_{q_{\bar{\alpha}2}}}\bar{\alpha}^2$$

$$C_N = (C_{n_{\bar{\alpha}}}\bar{\alpha} + C_{n_{\bar{\alpha}3}}\bar{\alpha}^3)\ \mathrm{Sin}\ (NF \cdot \phi) + (C_{n_{p_{\bar{\alpha}}}}\bar{\alpha} + C_{n_{p_{\bar{\alpha}3}}}\bar{\alpha}^3)\left(\frac{\dot{\phi}d}{2V_A}\right)$$

Individual coefficients are defined in the list of symbols.

In order to avoid ambiguities which might occur, the Euler angles ψ, θ and ϕ are limited to the following ranges:

$$-\pi < \psi < \pi$$

$$-\frac{\pi}{2} < \theta < \frac{\pi}{2}$$

$$0 < \phi < 2\pi$$

For non-planar motion, that is, pitching and yawing motions occurring simultaneously, the limits on the ranges of the Euler angles ψ and θ should be

$$-\frac{\pi}{6} < \psi < \frac{\pi}{6}$$

$$-\frac{\pi}{6} < \theta < \frac{\pi}{6}$$

to obtain reasonable accuracy of the coefficients extracted without an excessive number of iterations.

SECTION III

METHOD OF EXTRACTING COEFFICIENTS AND DESCRIPTION OF THE COMPUTER PROGRAM

1. Chapman and Kirk Coefficient Extraction Method

The value of using parametric influence coefficients in the analysis of dynamic systems has been recognized for some time. The following briefly describes the general scheme developed by Chapman and Kirk to use the method of parametric influence coefficients for determining aerodynamic coefficients. A more detailed presentation of the theory is given in references 1 and 2.

The basis of the method is the minimization of the deviations of a set of experimental data from a calculated motion. The system model that yields the calculated motion is given by the set of differential equations

$$\dot{x} = f(x,C,t) \quad , \quad x(0) = a \qquad (29)$$

where x(t) is an (n x 1) state vector, f is the (n x 1) vector-valued function, and a is the (n x 1) vector of initial conditions. The set of experimental data, z(t), are the components of the state vector. Assuming that x(t) is measured for $0 \leq t \leq \tau$, then it is desired to find the parameters C which minimize the expression

$$MSQE = \frac{1}{\tau} \int_0^\tau \{ x(t) - z(t) \}^T Q_w(t) \{ x(t) - z(t) \} dt \qquad (30)$$

where z(t) are the experimental data corresponding to the calculated motion of the state vector x(t) and $Q_w(t)$ is an (n x n) weighting matrix whose purpose is to give weight, or value, to only those components of the state vector x(t) for which measured experimental data are available.

The method used to determine the parameters C that satisfy equation (30) was an iterative one. For each iteration a calculated motion and a corresponding mean square error (MSQE) was determined. If the change in the root of the mean square error was not less than a predetermined value, the parameters C were updated, or corrected, toward that end. This was accomplished by integrating the set of equations obtained by taking the partial derivatives of each of the equations of motion with respect to the parameters of interest. These will be referred to as parametric differential equations in this paper. The solutions of the parametric differential equations, parameter influence coefficients, were then used to construct the (p x p) matrix of what will be referred to as parametric influence coefficients

$$A_{jk} = \sum_{i=1}^{NPTS} \left\{ \frac{\partial f}{\partial C_j} \right\}_i \left\{ \frac{\partial f}{\partial C_k} \right\}_i Q_w(t) \qquad (31)$$

For example, if there were 4 initial conditions and 6 coefficients, or 10 parameters of interest, sixty second order parametric differential equations were integrated to obtain the 10 x 10 matrix of parametric influence coefficients. Simultaneously, the gradient

$$B_j = \sum_{i=1}^{NPTS} \left\{ x(t) - z(t) \right\}_i \left\{ \frac{\partial f}{\partial C_j} \right\}_i Q_w(t) \qquad (32)$$

was obtained. Then the p x 1 matrix of parameter corrections, ΔC, was found by

$$[\Delta C] = [A]^{-1} [B] \qquad (33)$$

The parameters for the next, or $\ell + 1$, iteration were

$$[C]_{\ell + 1} = [C]_{\ell} + [\Delta C]_{\ell} \qquad (34)$$

Once the parameters were corrected, a new calculated motion was determined by integrating the equations of motion. The entire process was repeated until the predetermined value for the change of the root of the mean square error was satisfied, and the process was said to have converged, or the maximum number of iterations allowed was execeeded and the program was terminated.

2. Development of the Program

The program is written in Fortran IV for use primarily on an IBM 360/65 or 370/165 computer. The program provides the user with three general options:

1. Flight simulation
2. Coefficient extraction
3. Flight simulation with punched output of state vector component time histories.

The paragraphs that follow describe the functions of the main program, its subroutines, and the program options. A flow chart and a complete listing of the program and the required data input form are given in the appendices.

The function of the main program is to control the flow of the program in accordance with the options chosen. To do this, the main program reads and writes all input and output information, organizes the information, and calls the subroutines to use it. The main program does all of the calculations necessary to determine if convergence has been achieved and all of the calculations preparing for each iteration.

Subroutine ADDUM integrates the equations of motion and the parametric differential equations. The numerical method used is a fourth order Runge-Kutta starter solution and a fourth order Adams-Bashforth predictor-corrector method for integrating. This subroutine is described in detail in reference 5.

Subroutine XDOT1 computes current values of the derivatives of the set of first order equations to which the equations of motion have been reduced as required by ADDUM. The subroutine also computes the value of the derivative of the mean square error, which is integrated simultaneously with the equations of motion by ADDUM when the coefficient extraction option is specified.

Subroutine OUT1 stores the results of the numerically integrated equations of motion during each iteration until convergence is tested.

Subroutine XDOT2 computes current values of the derivatives of the set of first order equations to which the parametric differential equations have been reduced as required by ADDUM.

Subroutine OUT2 calculates the elements of the parametric influence coefficient matrix, [A], and the state vector difference matrix, [B], for use in the main program.

Subroutine MINV inverts the p x p matrix of parametric influence coefficients using a standard Gauss-Jordan method and is described in detail in reference 6.

Subroutine PLOT9 is a printer-plotter routine intended to give the program user a visual understanding of the angular orientation of the missile as calculated by the equations of motion.

The amount of input data required by the program is determined by the program option chosen. The specific data in each option are delineated in the following paragraphs. The formats and units of entries on specific data cards may be found in Appendix III.

(1) Flight simulation

(a) Program control codes

These integer constants tell the program which program options are in effect and which equations of motion are to have values computed for their derivatives in XDOT1. The integrated values of all other equations of motion are set to zero. The purpose of allowing the program user to specify the equations of motion that will have nonzero-integrated values is to avoid unnecessary computation, thus reducing execution time.

(b) Integration constants

The integration constants include the numerical integration step size, frequency of storage of integrated values, time at which integration is to stop, initial time, YES or NO codes to specify printer plots of each of the three angular motions, and the number of fins on the missile. The number of fins choice allows the user to specify a four-finned missile or an unfinned projectile.

(c) Aerodynamic and physical constants

The aerodynamic constants are air density and the free stream velocity that are specified during the flight simulation. The physical constants are the body cross-sectional area (neglecting fins), body diameter or equivalent, spin rate of the body at time zero, gravitational acceleration due to the earth, moment of inertia about the longitudinal axis, moment of inertia about the axes normal to the pitch and yaw planes, and mass of the body.

(d) Aerodynamic coefficients

For flight simulation the aerodynamic coefficients values are constant and are not altered by the program.

(e) Initial conditions

These values are the initial conditions for the equations of motion. Like the aerodynamic coefficients, they are constant and are not altered by the program.

(f) Printer plotter constants

These constants are required only if the plot option was specified in the integration constants. The constants are the width of the plot, value of the initial point, type of plot, field type for the data point values printed, and a scale factor.

(2) Coefficient extraction

(a) Output labels

These labels allow the program to identify the extracted values and the estimated standard deviations with appropriate labels.

(b) Program control codes

In addition to those listed for flight simulation, there are constants to specify initial conditions and aerodynamic coefficients to be adjusted, values for the weight factors in equation (30), maximum number of iterations allowed, and convergence tolerance before the iteration process is automatically terminated.

(c) Data

The initial condition and coefficient values input are now guesses and not constant values. In addition, values for the experimental data points of the state vector components must also be read.

(3) Flight simulation with punched output

The input for this option differs from the flight simulation only in the addition of a program control code to specify which state vector components are to be punched on cards.

SECTION IV

RESULTS OF TEST CASES

Eleven test cases of the program were run to check its operation. The test cases began with a one-degree-of-freedom case and were increased to a six-degree-of-freedom case. In all but two cases, the initial conditions and aerodynamic coefficients used to generate the data for the extraction program were known. This provided the easiest method for checking the validity of extracted initial conditions and coefficients. In the two cases where initial conditions and coefficients were not known, the extracted values were compared with those obtained from the same data by Daniel using UFPLANAR. The reason for investigating these two cases was to check the capability of the program to handle noisy data. The noise was simulated by random measurement errors in UFNOISE(2). A table of the results of the eleven cases may be found in Appendix IV.

The two cases with noisy data considered one-degree-of-freedom cases with linear and non-linear static restoring moment and pitch damping coefficients. As intuition would lead one to expect, the estimated standard deviations of the values extracted from noisy data were much larger than the standard deviations of the values extracted from data without noise. The standard deviations of the values extracted from data without noise were essentially zero, as they should have been, since the data were generated from the same equations of motion. However, the important result was that the number of iterations required for convergence was the same for both types of cases. This is very desirable from a computing standpoint because free flight test data will most certainly be noisy.

As stated previously in Chapter II, the mathematical model of the missile was restricted to low angles of attack for multiangular degree of freedom cases. In order to quantitatively demonstrate the necessity for this restriction, two cases were run with initial pitch and yaw angles both equal to 20 degrees in the first case and 30 degrees in the second. The 20-degree case required a reasonable six iterations to extract initial conditions and coefficients. On the other hand, the 30-degree case required eleven iterations to extract the correct values.

Several multi-degree-of-freedom combinations of angular and translational motions were among the cases run. It was found for these cases that the extraction process had to be a two- or three-step process, depending on the complexity of the case. The necessity for this procedure is a matter of the relative sensitivity of the parameters. This sensitivity may be observed by comparing the magnitudes of the elements along the main diagonal of the influence coefficient matrix. If a parameter is either insensitive or too sensitive to the motion of the missile, it will cause the adjustment of the parameters from iteration to iteration to be incorrect, that is, too small or too large. By carrying out the extraction process in a certain order of steps, this problem can be avoided. The steps should be as follows:

1. Extract initial conditions and coefficients related solely to translational motion.

2. Extract initial conditions and coefficients related solely to angular motion.

3. Extract coefficients related to interacting motions, such as the magnus forces and moments.

The order of the steps is as important as the steps themselves. The translational motion must be dealt with first since the formulation of the total angle of attack requires the inclusion of the velocity components.

For cases considering only pitching and/or yawing motions with a rolling motion, the extraction process requires only two steps:

1. Extract initial conditions and coefficients related solely to rolling motion.

2. Extract initial conditions and coefficients related to pitching and/or yawing motion.

It should be noted that the two aforementioned processes are recommendations.

SECTION V

CONCLUDING REMARKS

In summary, the purpose of this report was to construct a six-degree-of-freedom digital computer program which extracts aerodynamic coefficients from free flight test data using the Chapman and Kirk scheme. The mathematical model chosen for the program is somewhat arbitrary; the model has limitations such as the number and type of aerodynamic coefficients and magnitude of the angles of attack for multiangular degree of freedom cases as has already been shown. These limitations are not a function of the extraction scheme. With this in mind, the program was designed to be readily adaptable to a wide range of mathematical models. Major portions of the model are incorporated in subroutines to facilitate any changes. For instance, the input and output of information and the associated operations are contained in the main program, the model equations of motion are in XDOT1, and the parametric differential equations are in XDOT2. In addition to making program changes an easier process, this feature allows major portions of the program to be bypassed, depending on the program option chosen.

Segmenting the program into subroutines and using only those necessary in a given run is a method of keeping execution time to a minimum. However, since the program was designed to be capable of handling a maximum of six degrees of freedom, its operation on only one or two degrees of freedom is relatively costly. Thus, for maximum efficiency its use should be limited to multi-degree of freedom cases.

There are several areas of the program to be considered for further study or refinement. The first is the system model. One can see that this is an area of problem trade-offs. A very general model capable of handling a greater and more varied number of aerodynamic coefficients is more desirable from a purist's standpoint. However, the increased complexity and execution time of such a model is undesirable. Of course, the model may be designed to satisfy only certain requirements and yield good execution times but at the expense of generality. In addition, important aspects of any change are the time and effort necessary to make that change.

A second area that should be considered is the programming techniques used in constructing the program. The program was written in a straightforward manner rather like translating English to a foreign language word by word to make the program logic more understandable to the user. Although efficient programming techniques would reduce execution times, this is accompanied by a program that would be less understandable to the user.

Other areas that might be considered are much more complex. From informal discussions with Chapman and other sources, such as Meissinger(4), the author feels that the parameter influence coefficients and parametric influence coefficients in the [A] matrix are another important area for further study. It has already been found that the elements of the [A] matrix can be an important guide to the sensitivity of a parameter to the motion of the system model. With further study it might be possible to determine not only the numerical value of a parameter, but also its importance to the system relative to the other parameters in a quantitative sense rather than just a qualitative one.

(The reverse of this page is blank)

APPENDIX I

SIX-DEGREE-OF-FREEDOM NOMENCLATURE LIST AND PROGRAM LISTING

Nomenclature list (partial)

PROGRAM VARIABLE	MATH SYMBOL	DEFINITION
H	Δt	Numerical integration step size (sec)
ITO		Frequency of numerical integration output
TMAX	t_{max}	Cutoff time for numerical integration (sec)
TZERO	t_o	Initial time for numerical integration (sec)
XZO(I)		Initial condition labels
STDIC(I)		Initial condition standard deviation labels
COEF(I)		Coefficient labels
STDC(I)		Coefficient standard deviation labels
ICADJ(I)		Initial conditions extracted code
CADJ(I)		Coefficients extracted code
QW(I)	$Q_w(t)$	Weight factor
MAXIT		Maximum number of iterations before program terminates
TOL		Convergence criteria for change in root mean square error
NPTS		Number of experimental data points
N		Number of first order differential equations
RO	ρ	Air density

PROGRAM VARIABLE	MATH SYMBOL	DEFINITION
V	v	Wind tunnel velocity (ft/sec)
AR	A	Body reference area (ft)
D	d	Body reference diameter (ft)
P	p	Body spin rate (rad/sec)
G	g	Gravitational constant (ft/sec)
AIX	I_x	Moment of inertia about an axis longitudinally through the CG of the body (slug ft)
AM	m	Mass of the body (slugs)
CLA	$C_{\ell\bar{\alpha}}$	Rolling moment coefficient (rad)
CLP	$C_{\ell p}$	Roll damping coefficient (rad)
CMA	$C_{m\bar{\alpha}}$	Static pitching moment coefficient (rad)
CNA	$C_{n\bar{\alpha}}$	Pitching moment (due to fins) coefficient (rad)
CNPA	$C_{np\bar{\alpha}}$	Magnus moment coefficient (rad)
CMQO	C_{mqo}	Pitch and/or yaw damping coefficient (rad)
CXO	C_{xo}	Drag coefficient (rad)
CYA	$C_{y\bar{\alpha}}$	Side force coefficient (rad)
CYAP	$C_{y\bar{\alpha}p}$	Magnus force coefficient (rad)
CZA	$C_{z\bar{\alpha}}$	Normal force coefficient (rad)
$(\ )^2$	$(\ )^2$	Second order term
$(\ )^3$	$(\ )^3$	Third order term
		Angle of pitch in X-Z plane (rad)

PROGRAM VARIABLE	MATH SYMBOL	DEFINITION
		Angle of yaw in X-Y plane (rad)
		Angle of body roll relative to fixed plane axis system (rad)
X	x	X position of body relative to tunnel reference point (ft)
Y	y	Y position of body relative to tunnel reference point (ft)
Z	z	Z position of body relative to tunnel reference point (ft)
( )	$\frac{d}{dt}$	First derivative with respect to time
( )	$\frac{d^2}{dt^2}$	Second derivative with respect to time
DATUM(I)	z(t)	Experimental data point values
DCALC(I)	x(t)	Calculated point values
IEQ(I)		Equations of motion to be integrated code
IP		Psi plot code
IT		Theta plot code
IFE		Phi plot code
NF		Number of fins code

Program Listing

```
      DIMENSION XZ(372),TI(500),XZERO(372),
     1STDC(19),SDCO(19),ISV(12)
      DIMENSION DELC(31),STDEV(31),CEXT(19),XZO(12),
     1STDIC(12),COEF(19)
      DIMENSION AJK1(961),L1(31),L2(31)
      INTEGER TEST1,TEST2,CADJ(19)
      DIMENSION Y(200,1),X(2232)
      REAL*4 FMT(6)
      REAL MSQE
      COMMON N,T,X
      COMMON V,RO,AR,D,AIX,AI,AM,G,P
      COMMON DATUM(12,500),DCALC(12,500)
      COMMON /DATA1/NEQ,NIC,NC,NP,CADJ,QW(12),AK1,NPTS,KKK,
     1C(19),H,B(31),IEQ(6),MODE,JJJ,NF,ICADJ(12),JT
      COMMON /DATA2/AJK(31,31)
      EXTERNAL XDOT1,OUT1,XDOT2,OUT2
      N=12
      NPL=1
      JJJ=0
      READ(5,103)MODE
103   FORMAT(I1)
      KWR=MODE+1
      GO TO (1031,1032,1033),KWR
1031  WRITE(6,2031)
      GO TO 1035
2031  FORMAT('0',5X,'THE PURPOSE OF THIS RUN IS FLIGHT SIMULATION')
1032  WRITE(6,2032)
2032  FORMAT('0',5X,'THE PURPOSE OF THIS RUN IS COEFFICIENT',
     1' EXTRACTION.')
      GO TO 1034
1033  WRITE(6,2033)
2033  FORMAT('0',5X,'THE PURPOSE OF THIS RUN IS FLIGHT ',
     1'SIMULATION WITH PUNCHED OUTPUT.')
      GO TO 1035
```

```
1034   READ(5,1002)(XZO(I),I=1,12)
       READ(5,1002)(STDIC(I),I=1,12)
1002   FORMAT(12A4)
       READ(5,1003)(COEF(I),I=1,19)
       READ(5,1003)(STDC(I),I=1,19)
1003   FORMAT(19A4)
       READ(5,101)(ICADJ(KIC),KIC=1,12)
101    FORMAT(12I1)
       READ(5,102)(CADJ(KC),KC=1,19)
102    FORMAT(19I1)
       READ(5,104)(QW(KWF),KWF=1,12)
104    FORMAT(12F5.2)
       READ(5,3)MAXIT,TOL
3      FORMAT(I2,E10.4)
       WRITE(6,33)MAXIT,TOL
33     FORMAT('C',5X,'IF THE SOLUTION DOES NOT CONVERGE IN',
      1I2,' ITERATIONS OR SATISFY THE CONVERGENCE TOLERANCE',
      2E12.5,' THE PROGRAM TERMINATES.')
C      DETERMINE NUMBER AND WHICH INITIAL CONDITIONS ARE TO BE
C      ADJUSTED
       NIC=0
       DO 110 KIC=1,12
       TEST1=ICADJ(KIC)+1
       GO TO (110,111),TEST1
111    NIC=NIC+1
       ICADJ(NIC)=KIC
110    CONTINUE
C      NIC IS THE NUMBER OF INITIAL CONDITIONS TO BE ADJUSTED
C      ICADJ CONTAINS POINTERS TO IC'S TO BE ADJUSTED
C      NOW DETERMINE NUMBER AND WHICH COEFFICIENTS ARE TO BE ADJUSTED
       NC=0
       DO 120 KC=1,19
       TEST2=CADJ(KC)+1
       GO TO (120,121),TEST2
```

Copy available to DDC does not permit fully legible reproduction

```
121    NC=NC+1
       CADJ(NC)=KC
120    CONTINUE
C      NC IS NUMBER OF COEFFICIENTS TO BE ADJUSTED
C      CADJ CONTAINS POINTERS TO THOSE COEFFICIENTS
1035   READ(5,1001)(IEQ(KEQ),KEQ=1,6)
1001   FORMAT(6I1)
C      DETERMINE NUMBER AND WHICH EQUATIONS OF MOTION ARE TO BE
C      INTEGRATED
       NEQ=0
       DO 112 KEQ=1,6
       JTEST=IEQ(KEQ)+1
       GO TO (112,113),JTEST
113    NEQ=NEQ+1
       IEQ(NEQ)=KEQ
112    CONTINUE
C      NEQ IS THE NUMBER OF EQUATIONS TO BE INTEGRATED
C      IEQ CONTAINS POINTERS TO EQUATIONS TO BE INTEGRATED
       WRITE(6,2001) NEQ
2001   FORMAT('0',5X,'THERE ARE ',I1,' DEGREES OF FREEDOM.')
       IF(MODE.EQ.0) GO TO 1036
       READ(5,101)(ISV(KSV),KSV=1,12)
1036   READ(5,1)H,ITO,TMAX,TZERO,IP,IT,IFE,NF
1      FORMAT(F5.3,I3,2F5.3,4I1)
       WRITE(6,2)H,ITO,TMAX,TZERO,N,IP,IT,IFE,NF
2      FORMAT('0',5X,'INTEGRATION CONSTANTS'/'+',5X,21('  ')/
      1'0',10X,'H=',F5.3,2X,'ITO=',I3,2X,'TMAX=',F5.3,2X,'TZERO=',
      2F5.3,2X,'N=',I2,2X,'IP=',I1,2X,'IT=',I1,2X,'IFE=',I1,
      22X,3HNF=,I2)
       ATO=ITO
       NPTS=((TMAX-TZERO)/(H*ATO))+1.2
       IF(MODE.EQ.0) GO TO 1161
C      DETERMINE NUMBER AND WHICH STATE VECTORS ARE TO BE READ
C      AND INFLUENCING THE RMS ERROR
```

```
      NSV=0
      DO 116 KSV=1,12
      LTEST=ISV(KSV)+1
      GO TO (116,117),LTEST
117   NSV=NSV+1
      ISV(NSV)=KSV
116   CONTINUE
C     NSV IS THE NUMBER OF STATE VECTORS AND THE ISV'S
C     CONTAINS POINTERS
1161  IF(MODE.NE.1) GO TO 130
      WRITE(6,201)(XZO(ICADJ(I)),I=1,NIC)
201   FORMAT(1HO,5X,35HTHE FOLLOWING INITIAL CONDITIONS ARE,
     122H ADJUSTED IN THIS RUN /1HO,10X,12(A4,2X))
      WRITE(6,202)(COEF(CADJ(I)),I=1,NC)
202   FORMAT('0',5X,'THE FOLLOWING COEFFICIENTS ARE ADJUSTED',
     1' IN THIS RUNO'/'0',10X,15(A4,2X))
      WRITE(6,2051)
2051  FORMAT('0',5X,'TIME HISTORIES OF THE FOLLOWING STATE',
     1' VECTOR COMPONENTS AND THEIR TIME DERIVATIVES ARE INPUT',
     2' AS DATA'/5X,'WITH THE WEIGHTING FACTORS SHOWN FOR',
     3' THE MEAN SQUARED ERROR CALCULATION.')
      WRITE(6,2052)(XZO(ISV(I)),QW(ISV(I)),I=1,NSV)
2052  FORMAT('0',10X,A4,' 0 ',F5.2)
      NP=NC+NIC
      NALL=NP*12
      DO 122 NA=1,NALL
122   XZERO(NA)=0.0
      IF(NIC)1241,1241,123
123   DO 124 NK=1,NIC
      J=ICADJ(NK)+(NK-1)*12
      XZERO(J)=1.0
124   CONTINUE
1241  CONTINUE
C     START FLYSIM
```

```
130    READ(5,106)RO,V,AR,D,P,G,AIX,AI,AM
106     FORMAT(F11.8,4F11.4)
        WRITE(6,12)
12      FORMAT('O',10X,'AERODYNAMIC AND PHYSICAL INPUT DATA'/'+',10X,
       135(' '))
        WRITE(6,206)RO,V,AR,D,P,G,AIX,AI,AM
206     FORMAT(1HO,9X,2HRO,11X,1HV,10X,4HAREA,6X,8HDIAMETER,4X,
       19HSPIN-RATE/1H ,5X,11H(SLG/FT**3),3X,8H(FT/SEC),5X,7H(FT**2),
       21X,4H(FT).6X,9H(RAD/SEC),/1HO,5X,F10.7,3X,F9.3,2(3X,F9.5),3X,
       3F10.3//7X,7HGRAVITY,8X,2HIX,11X,1HI,9X,4HMASS/5X,
       411H(FT/SEC**2),2X,11H(SLG-FT**2),1X,11H(SLG-FT**2),3X,
       56H(SLGS),//6X,F10.6,3(3X,F9.5))
        READ(5,107)CLA,CLA3,CLP,CMA,CMA3,CNA,CNA3,CNPA,CNPA3,CMQO,
       1CMQ2,CXO,CXA2,CYA,CYA3,CYAP,CYAP3,CZA,CZA3
107     FORMAT(8F10.4)
        WRITE(6,14)
14      FORMAT('O',10X,'AERODYNAMIC COEFFICIENT ESTIMATES'/'+',10X,
       133(' '))
        WRITE(6,207)CLA,CLA3,CLP,CMA,CMA3,CNA,CNA3,CNPA,CNPA3,
       1CMQO,CMQ2,CXO,CXA2,CYA,CYA3,CYAP,CYAP3,CZA,CZA3,NPTS
207     FORMAT('O',5X,'CLA=',F7.3,4X,'CLA3=',F7.3,3X,'CLP=',F7.3,4X,
       1'CMA=',F7.3,4X,'CMA3=',F7.3,3X,'CNA=',F7.3,4X,'CNA3=',F7.3/5X,
       2'CNPA=',F7.3,3X,'CNPA3=',F7.3,2X,'CMQO=',F8.3,3X,'CMQ2=',F8.3,3X,
       3'CXO=',F7.3,4X,'CXA2=',F7.3,3X,'CYA=',F7.3,/5X,'CYA3=',F7.3,3X,
       4'CYAP=',F7.3,3X,'CYAP3=',F7.3,2X,'CZA=',F7.3,4X,'CZA3=',F7.3,
       53X,'NPTS=',I3)
        AK1=P*AIX/AI
C       MODEO 0= FLIGHT SIMULATION
C             1= READ DATA AND EXTRACT COEFFICIENTS
C             2= FLIGHT SIMULATION WITH PUNCHED OUTPUT
        KTEST=MODE+1
        GO TO (139,131,139),KTEST
131     DO 1311 I=1,NSV
1311    READ(5,105)(DATUM(ISV(I),KPT),KPT=1,NPTS)
```

```
105    FORMAT(5E16.6)
       DO 1312 I=1,NSV
1312   WRITE(6,205)(DATUM(ISV(I),KPT),KPT=1,NPTS)
205    FORMAT(1X,5E16.6)
       N=N+1
139    CONTINUE
       READ(5,108)(XZ(I),I=1,N)
108    FORMAT(8F10.4)
       VA=SQRT((V+XZ(8))**2+XZ(10)**2+XZ(12)**2)
       Q=0.5*RO*VA**2
       CON1=(Q*AR*D)/AIX
       CON3=(Q*AR*D)/AI
       CON5=Q*AR*AM
       IF(VA.EQ.0) GO TO 999
       CON2=(Q*AR*D**2)/(2*VA*AIX)
       CON4=(Q*AR*D**2)/(2*VA*AI)
       C(1)=CON1*CLA
       C(2)=CON1*CLA3
       C(3)=CON2*CLP
       C(4)=CON3*CMA
       C(5)=CON3*CMA3
       C(6)=CON3*CNA
       C(7)=CON3*CNA3
       C(8)=CON4*CNPA
       C(9)=CON4*CNPA3
       C(10)=CON4*CMQ0
       C(11)=CON4*CMQ2
       C(12)=CON5*CX0
       C(13)=CON5*CXA2
       C(14)=CON5*CYA
       C(15)=CON5*CYA3
       C(16)=CON4*CYAP
       C(17)=CON4*CYAP3
       C(18)=CON5*CZA
```

```
      C(19)=CON5*CZA3
1391  CONTINUE
      KKK=0
140   CALL ADDUM(H,ITO,TZERO,TMAX,XZ,XDOT1,OUT1,+999)
      IF(N.LT.13) GO TO 1429
      MSQE=X(13)
1429  KTEST=MODE+1
      GO TO (141,142,143),KTEST
143   DO 1431 I=1,NSV
1431  WRITE(7,105)(DCALC(ISV(I),KPT),KPT=1,NPTS)
      GO TO 141
142   DO 144 J=1,NP
      B(J)=0.0
      DO 144 K=1,NP
144   AJK(J,K)=0.0
      N=(NIC+NC)*12
      CALL ADDUM(H,ITO,TZERO,TMAX,XZERO,XDOT2,OUT2,+999)
      IF(JJJ.GT.0) GO TO 171
      WRITE(6,270)
270   FORMAT('1',5X,'CURRENT PARAMETER VALUES AREO')
      GO TO 172
171   WRITE(6,271)
271   FORMAT('0',5X,'CURRENT PARAMETER VALUES AREO')
172   WRITE(6,272)JJJ,(XZ(I),I=1,12)
272   FORMAT('0',5X,'ITERATION NUMBERO',I2/1HO,5X,'PSI=',F10.4,
     12X,'PSIDOT=',F10.4,2X,'THA=',F10.4,2X,'THADOT=',F10.4,2X,
     2'FEE=',F10.4,2X,'FEEDOT=',F10.4,2X/'0',5X,'EX=',F10.4,2X,
     3'EXDOT=',F10.4,2X,'WY=',F10.4,2X,'WYDOT=',F10.4,2X,'ZE=',
     4F10.4,2X,'ZEDOT=',F10.4)
      WRITE(6,273) (C(I),I=1,19)
273   FORMAT('0',5X,'C13=',E12.5,5X,'C14=',E12.5,5X,'C15=',E12.5,
     15X,'C16=',E12.5,5X,'C17=',E12.5/'0',5X,'C18=',E12.5,5X,
     2'C19=',E12.5,5X,'C20=',E12.5,5X,'C21=',E12.5,5X,'C22=',
     3E12.5/'0',5X,'C23=',E12.5,5X,'C24=',E12.5,5X,'C25=',E12.5,
     45X,'C26=',E12.5,5X,'C27=',E12.5/'0',5X,'C28=',E12.5,5X,
     5'C29=',E12.5,5X,'C30=',E12.5,5X,'C31=',E12.5)
      DO 173 J=1,NP
      DELC(J)=0.0
173   STDEV(J)=0.0
      DO 590 M=1,NP
      DO 590 N=1,NP
590   AJK1((M-1)*NP+N)=AJK(M,N)
      CALL MINV(AJK1,NP,R,L1,L2)
      DO 591 M=1,NP
      DO 591 N=1,NP
591   AJK(M,N)=AJK1((M-1)*NP+N)
      IF(JJJ.GT.0) GO TO 175
      RMSE=0.0
      DIFF=1.0E20
175   RMSEP=RMSE
      DIFFP=DIFF
      RMSE=SQRT(MSQE/(TMAX-TZERO))
      DIFF=ABS(RMSE-RMSEP)
1751  DO 176 J=1,NP
176   STDEV(J)=RMSE*SQRT(AJK(J,J))
      WRITE(6,274)RMSE
274   FORMAT('0',5X,'RMS ERROR=',E16.8)
      WRITE(6,2741)
2741  FORMAT('0',10X,'CURRENT PARAMETER STANDARD DEVIATIONS',
     1' AREO'/'+',10X,42(' '))
      WRITE(6,275)(ICADJ(I),STDEV(I),I=1,NIC)
275   FORMAT('0',5X,'SDIC(',I2,')=',E12.5,5X,'SDIC(',I2,')=',
     1E12.5,5X,'SDIC(',I2,')=',E12.5,5X,'SDIC(',I2,')=',E12.5)
      WRITE(6,276)(CADJ(I),STDEV(I+NIC),I=1,NC)
276   FORMAT('0',5X,'SDC(',I2,')=',E12.5,5X,'SDC(',I2,')=',E12.5,
     15X,'SDC(',I2,')=',E12.5,5X,'SDC(',I2,')=',E12.5)
      IF(DIFF.GT.TOL) GO TO 1991
      IF(MSQE.LT.SQRT(TOL)) GO TO 199
```

```
1991  IF(JJJ.GE.MAXIT) GO TO 197
      JJJ=JJJ+1
      DO 177 I=1,NP
      DO 177 J=1,NP
177   DELC(I)=DELC(I)+AJK(I,J)*B(J)
      WRITE(6,277)
277   FORMAT('0',5X,'CURRENT INITIAL CONDITION CORRECTIONS AREO')
      WRITE(6,278)(ICADJ(I),DELC(I),I=1,NIC)
278   FORMAT('C',5X,'DELIC(',I2,')=',E12.5,5X,'DELIC(',I2,')=',
     1E12.5,5X,'DELIC(',I2,')=',E12.5,5X,'DELIC(',I2,')=',E12.5)
      WRITE(6,279)
279   FORMAT('0',5X,'CURRENT COEFFICIENT CORRECTIONS AREO')
      WRITE(6,280)(CADJ(I),DELC(I+NIC),I=1,NC)
280   FORMAT('0',5X,'DELC(',I2,')=',E12.5,5X,'DELC(',I2,')=',
     1E12.5,'DELC(',I2,')=',E12.5,5X,'DELC(',I2,')=',E12.5)
      GO TO 198
197   WRITE(6,281)
281   FORMAT('0',5X,'CONVERGENCE FAILED - MAXIMUM NUMBER OF',
     .11X,' ITERATIONS EXCEEDED')
      GO TO 9999
198   DO 178 KIC=1,NIC
      J=ICADJ(KIC)
178   XZ(J)=XZ(J)+DELC(KIC)
      DO 179 KC=1,NC
      K=KC+NIC
      J=CADJ(KC)
179   C(J)=C(J)+DELC(K)
      N=13
      GO TO 1991
199   VA=SQRT((V+XZ(8))**2+XZ(10)**2+XZ(12)**2)
      Q=0.5*RO*VA**2
      CON1=(Q*AR*D)/AIX
      CON3=(Q*AR*D)/AI
      CON5=Q*AR*AM
      IF(VA.EQ.0) GO TO 999
      CON2=(Q*AR*D**2)/(2*VA*AIX)
      CON4=(Q*AR*D**2)/(2*VA*AI)
      DO 190 I=1,NC
      MC=CADJ(I)
      GO TO (191,191,192,193,193,193,193,194,194,194,194,195,195,
     1195,195,194,194,195,195),MC
  191 CEXT(MC)=C(MC)/CON1
      GO TO 190
  192 CEXT(MC)=C(MC)/CON2
      GO TO 190
  193 CEXT(MC)=C(MC)/CON3
      GO TO 190
  194 CEXT(MC)=C(MC)/CON4
      GO TO 190
  195 CEXT(MC)=C(MC)/CON5
  190 CONTINUE
      DO 180 I=1,NC
      M=CADJ(I)
      GO TO (181,181,182,183,183,183,183,184,184,184,184,185,185,
     1185,185,184,184,185,185),M
  181 SDCO(M)=STDEV(NIC+I)/CON1
      GO TO 180
  182 SDCO(M)=STDEV(NIC+I)/CON2
      GO TO 180
  183 SDCO(M)=STDEV(NIC+I)/CON3
      GO TO 180
  184 SDCO(M)=STDEV(NIC+I)/CON4
      GO TO 180
  185 SDCO(M)=STDEV(NIC+I)/CON5
  180 CONTINUE
      WRITE(6,284)
  284 FORMAT('0',5X,'EXTRACTED INITIAL CONDITIONS AND',
     1' THEIR STANDARD DEVIATIONS ARE0')
```

```
      WRITE(6,285)(XZ0(ICADJ(I)),XZ(ICADJ(I)),I=1,NIC)
285   FORMAT('0',4(5X,A4,'=',F9.4))
      WRITE(6,282)(STDIC(ICADJ(I)),STDEV(I),I=1,NIC)
282   FORMAT('0',4(5X,A4,'=',E12.5))
      WRITE(6,286)
286   FORMAT('0',5X,'EXTRACTED COEFFICIENTS AND THEIR',
     1' STANDARD DEVIATIONS ARE ')
      WRITE(6,287)(COEF(CADJ(I)),CEXT(CADJ(I)),I=1,NC)
287   FORMAT('0',3(5X,A4,'=',F9.3))
      WRITE(6,288)(STDC(CADJ(I)),SDCO(CADJ(I)),I=1,NC)
288   FORMAT('0',5(5X,A4,'=',E12.5))
141   CONTINUE
      DO 132 I=1,NPTS
      TI(I)=H*ITO*(I-1)
132   CONTINUE
      NOUT=1
152   IF(NPTS.LE.(NOUT+49)) GO TO 155
      WRITE(6,215)
215   FORMAT('1',7X,'TIME',7X,'ANGLE-OF-YAW',4X,'ANGLE-OF-PITCH',
     15X,'ANGLE-OF-ROLL',5X,'X-POSITION',5X,'Y-POSITION',5X,
     2'Z-POSITION',/5X,'(SECONDS)',7X,'(RADIANS)',8X,'(RADIANS)',
     39X,'(RADIANS)',9X,'(FEET)',9X,'(FEET)',9X,'(FEET)'/)
      NUP=NOUT+49
      DO 211 I=NOUT,NUP
211   WRITE(6,216)TI(I),(DCALC(KX,I),KX=1,11,2)
216   FORMAT(5X,F8.5,7X,F8.5,9X,F8.5,10X,F8.5,9X,F8.3,2(7X,F8.3))
      NOUT=NOUT+50
      GO TO 152
155   WRITE(6,215)
      DO 220 I=NOUT,NPTS
220   WRITE(6,216)TI(I),(DCALC(KX,I),KX=1,11,2)
      TIME=H*ITO
      IF(IP.EQ.1) GO TO 301
302   IF(IT.EQ.1) GO TO 303
```

```
304   IF(IFE.EQ.1) GO TO 307
      GO TO 9999
301   DO 305 K=1,200
305   Y(K,1)=DCALC(1,K)+0.6
      WRITE (6,310)
310   FORMAT(1H1//55X,8HPSI PLOT)
      WRITE(6,320)TIME
320   FORMAT(1H0,46X,28HTIME BETWEEN DATA POINTS IS ,F5.3,3HSEC,//)
      WRITE(6,322)
322   FORMAT(1X,'-0.6',6X,'-0.5',6X,'-0.4',6X,'-0.3',6X,'-0.2',
     16X,'-0.1',7X,'0.0',7X,'0.1',7X,'0.2',7X,'0.3',7X,'0.4',7X,
     2'0.5',7X,'0.6',//)
      CALL PLOT9(Y,K,NPL)
      GO TO 302
303   DO 306 K=1,200
306   Y(K,1)=DCALC(3,K)+0.6
      WRITE(6,311)
311   FORMAT(1H1//55X,10HTHETA PLOT)
      WRITE(6,320)TIME
      WRITE(6,322)
      CALL PLOT9(Y,K,NPL)
      GO TO 304
307   DO 308 K=1,200
308   Y(K,1)=DCALC(5,K)+6.0
      WRITE(6,312)
312   FORMAT(1H1//55X,8HPHI PLOT)
      WRITE(6,320)TIME
      WRITE(6,323)
323   FORMAT(1X,'-6.0',6X,'-5.0',6X,'-4.0',6X,'-3.0',6X,'-2.0',
     16X,'-1.0',7X,'0.0',7X,'1.0',7X,'2.0',7X,'3.0',7X,'4.0',7X,
     2'5.0',7X,'6.0',//)
      CALL PLOT9(Y,K,NPL)
 9998 GO TO 9999
997   WRITE(6,998)
```

```
998   FORMAT('0',5X,'THE RMS ERROR IS GETTING WORSE. TRY ',
     1'BETTER GUESSES OR MORE DATA POINTS')
      GO TO 9999
999   WRITE(6,115)
115   FORMAT(1H0,5X,'THERE WAS A DIVISION BY ZERO THAT RESULTED IN ',
     1'INFINITY')
9999  CONTINUE
      STOP
      END
      SUBRCUTINE ADDUM(AITCH,ITO,TZERO,TMAX,XZ,F,OUT,*)
      DIMENSION XZ(372),CX(372,6),X(372,6)
      COMMON N,T,X
      H = ABS(AITCH)                                                  ADDU 120
      IT =ITO-1                                                       ADDU 130
      D=TZERO-TMAX                                                    ADDU 140
      IF (D)2,1,1                                                     ADDU 150
    1 H=-H                                                            ADDU 160
    2 HH=0.5*H                                                        ADDU 170
      C=H/24.0                                                        ADDU 180
      T=TZERO                                                         ADDU 190
      ISET=0                                                          ADDU 200
      DO        3 I=1,N                                               ADDU 210
      X(I,1)=XZ(I)                                                    ADDU 220
    3 X(I,6)=XZ(I)                                                    ADDU 230
      CALL F(X(1,5),1,+99)
      CALL F(X(1,2),0,+99)
      IF(ITO)5,5,4                                                    ADDU 260
4            CALL OUT(1)
    5 DO        6 K=1,N                                               ADDU 280
           CX(K,1) = X(K,5)*HH
    6      X(K,1) = X(K,1) + CX(K,1)
      T=T+HH                                                          ADDU 310
      CALL F(CX(1,2),3,+99)
      DO        7 K=1,N                                               ADDU 330
           CX(K,2) = HH*CX(K,2)
    7      X(K,1) = X(K,6) + CX(K,2)
      CALL F(CX(1,3),3,+99)
      DO        8 K=1,N                                               ADDU 370
           CX(K,3) = H*CX(K,3)
    8      X(K,1) = X(K,6) + CX(K,3)
      T=T+HH                                                          ADDU 400
      CALL F(CX(1,4),1,+99)
      DO        9 K=1,N                                               ADDU 420
      CX(K,4) = CX(K,4)*H
      X(K,1)=X(K,6)+(CX(K,1)+2.0*CX(K,2)+CX(K,3)+
     1 0.5*CX(K,4))*0.333333333333
  9   X(K,6) = X(K,1)                                           ADDU 450
    ISET = ISET+1                                               ADDU 460
    GO TO (10,12,17),ISET                                       ADDU 470
 10 CALL F(X(1,5),0,+99)
    DO     11 K = 1,N                                           ADDU 490
 11 X(K,3)=X(K,5)                                               ADDU 500
       GO TO     18                                             ADDU 510
 12 CALL F(X(1,5),0,+99)
    DO     13 K=1,N                                             ADDU 530
 13 X(K,4)=X(K,5)                                               ADDU 540
       GO TO     18                                             ADDU 550
 14 T=T+H                                                       ADDU 560
    DO     15 K=1,N                                             ADDU 570
    X(K,1)=X(K,6)+D*(55.0*X(K,5)-59.0*X(K,4)+37.0*X(K,3)-9.0*X(K,2))
    X(K,2)=X(K,3)                                               AD U 5 0
    X(K,3)=X(K,4)                                               ADDU 600
 15 X(K,4)=X(K,5)                                               ADDU 610
    CALL F(X(1,5),1,+99)
    DO     16 K = 1,N                                           ADDU 630
       X(K,6)=X(K,6)+D*(9.0*X(K,5)+19.0*X(K,4)-5.0*X(K,3)+X(K,2))  ADDU 640
 16    X(K,1)=X(K,6)
    ISET = 3                                                    ++ 4
 17 CALL F(X(1,5),0,+99)
 18 IF (IT)20,19,20                                              4
 19    IT = ITO                                                 ADDU 690
       CALL OUT(0)
 20 IT = IT - 1                                                 ADDU 710
    IF(ABS(TMAX-T)-ABS(HH))21,21,22                             ADDU 720
 21   RETURN                                                    ADDU 730
 22   GO TO (5,5,14),ISET
99  RETURN 1
    END                                                         ADDU 750
```

```
      SUBROUTINE XDOT1(A,K,*)
      INTEGER CADJ(19)
      DIMENSION A(13),XZ(380)
      COMMON N,T,X(2232)
      COMMON V,RO,AR,D,AIX,AI,AM,G,P
      COMMON DATUM(12,500),DCALC(12,500)
      COMMON /DATA1/NEQ,NIC,NC,NP,CADJ,QW(12),AK1,NPTS,KKK,
     1C(19),H,B(31),IEQ(6),MODE,JJJ,NF,ICADJ(12),JT
      EQUIVALENCE (X(1),PSI),(X(2),PSIDOT),(X(3),THA),(X(4),THADOT),
     1(X(5),FEE),(X(6),FEEDOT),(X(7),EX),(X(8),EXDOT),(X(9),WY),
     2(X(10),WYDOT),(X(11),ZE),(X(12),ZEDOT)
      DO 2 I=1,12
2     A(I)=0.0
747   ST=SIN(THA)
      CT=COS(THA)
      SP=SIN(PSI)
      CP=COS(PSI)
92    R1=(V+EXDOT)*ST*CP+WYDOT*ST*SP+ZEDOT*CT
      R2=-(V+EXDOT)*SP+WYDOT*CP
      R3=(V+EXDOT)*CT*CP+WYDOT*CT*SP-ZEDOT*ST
      IF(R3.EQ.0) GO TO 6
      ARGA=R1/R3
      ARGB=R2/R3
      IERROR=1
      GO TO 13
6     IF(R1.EQ.0) GO TO 7
      IERROR=2
      GO TO 100
7     ARGA=0
      IERROR=1
11    IF(R2.EQ.0) GO TO 12
      IERROR=2
      GO TO 100
12    ARGB=0
```

```
      IERROR=1
13    AHAT=ATAN(ARGA)
      BHAT=ATAN(ARGB)
      ALBAR=SQRT(AHAT**2+BHAT**2)
      SA=SIN(AHAT)
      SB=SIN(BHAT)
      SAL=SIN(ALBAR)
      IF(ALBAR.EQ.0) GO TO 601
      CSFEHT=SA/SAL
      SNFEHT=SB/SAL
      GO TO 602
601   CSFEHT=1.0
      SNFEHT=0.0
602   FEEP=FEE+ARSIN(SNFEHT)
      S4F=SIN(NF*FEEP)
      CC1=C(1)*ALBAR+C(2)*ALBAR**3
      CC2=C(4)*ALBAR+C(5)*ALBAR**3
      CC3=C(6)*ALBAR+C(7)*ALBAR**3
      CC4=C(8)*ALBAR+C(9)*ALBAR**3
      CC5=C(10)+C(11)*ALBAR**2
      CC6=C(12)+C(13)*ALBAR**2
      CC7=C(14)*ALBAR+C(15)*ALBAR**3
      CC8=C(16)*ALBAR+C(17)*ALBAR**3
      CC9=C(18)*ALBAR+C(19)*ALBAR**3
      DO 1000 I=1,NEQ
      KT=IEQ(I)
      GO TO (21,22,23,24,25,26),KT
21    A(1)=PSIDOT
      A1=(AK1+2*A(1)*ST)*THADOT/CT
      A2=CC3*S4F*CSFEHT
      A3=CC4*FEEDOT*CSFEHT
      A4=-CC2*SNFEHT
      A5=CC5*A(1)
      A(2)=A1+(A2+A3+A4)/CT+A5
      GO TO 1000
22    A(3)=THADOT
      A6=-AK1*A(1)*CT-A(1)**2*ST*CT
      A7=CC2*CSFEHT
      A8=(CC3*S4F+CC4*FEEDOT)*SNFEHT
      A9=CC5*THADOT
      A(4)=A6+A7+A8+A9
      GO TC 1000
23    A(5)=FEEDOT+P
      A(6)=CC1*S4F+C(3)*A(5)+A(1)*A(3)*CT+ST*X(1)
      GO TO 1000
24    A(7)=EXDOT
      A13=CC6*CT*CP
      A14=CC7*S4F
      A15=CC8*A(5)
      A16=SNFEHT*ST*CP+CSFEHT*SP
      A17=CC9
      A18=-CSFEHT*ST*CP+SNFEHT*SP
      A(8)=A13-(A14+A15)*A16-A17*A18
      GO TO 1000
25    A(9)=WYDOT
      A19=CC6*CT*SP
      A20=CSFEHT*CP-SNFEHT*ST*SP
      A21=CSFEHT*ST*SP+SNFEHT*CP
      A(10)=A19+(A14+A15)*A20+A17*A21
      GO TO 1000
26    A(11)=ZEDOT
      A22=-CC6*ST
      A23=SNFEHT*CT
      A24=CSFEHT*CT
      A(12)=A22-(A14+A15)*A23+A17*A24-G
1000  CONTINUE
      J=T/H+1.2
      IF(MODE.NE.1) GO TO 70
```

```
40     DMSE=0.0
       DO 60 I=1,12
       IF(QW(I))61,60,61
61     IF(K-3)62,63,62
63     DMSE=QW(I)*(X(I)-.5*DATUM(I,J)-.5*DATUM(I,J+1))**2+DMSE
       GO TO 60
62     DMSE=QW(I)*(X(I)-DATUM(I,J))**2+DMSE
60     CONTINUE
 65    A(13)=DMSE
100    GO TO(70,80),IERROR
70     RETURN
80     RETURN 1
       END
```

```
      SUBROUTINE OUT1(K)
      INTEGER CADJ(19)
      COMMON N,T,X(2232)
      COMMON V,RO,AR,D,AIX,AI,AM,G,P
      COMMON DATUM(12,500),DCALC(12,500)
      COMMON /DATA1/NEQ,NIC,NC,NP,CADJ,QW(12),AK1,NPTS,KKK,
     1C(19),H,B(31),IEQ(6),MODE,JJJ,NF,ICADJ(12),JT
      KKK=KKK+1
      DO 700 I=1,12
700   DCALC(I,KKK)=X(I)
      RETURN
      END
      SUBROUTINE XDOT2(DOT,KDUM,*)
      INTEGER CADJ(19)
      DIMENSION DOT(1),FSX(12,12)
      COMMON N,T,X(2232)
      COMMON V,RO,AR,D,AIX,AI,AM,G,P
      COMMON DATUM(12,500),DCALC(12,500)
      COMMON /DATA1/NEQ,NIC,NC,NP,CADJ,CW(12),AK1,NPTS,KKK,
     1C(19),H,B(31),IEQ(6),MODE,JJJ,NF,ICADJ(12),JT
      DATA FSX/144*0.0/
      IF(KDUM.NE.1) GO TO 10
1     JT=T/H+1.5
      PSI=DCALC(1,JT)
      PSIDOT=DCALC(2,JT)
      THA=DCALC(3,JT)
      THADOT=DCALC(4,JT)
      FEE=DCALC(5,JT)
      FEEDOT=DCALC(6,JT)
      EX=DCALC(7,JT)
      EXDOT=DCALC(8,JT)
      WY=DCALC(9,JT)
      WYDOT=DCALC(10,JT)
      ZE=DCALC(11,JT)
      ZEDOT=DCALC(12,JT)
747   SP=SIN(PSI)
      CP=COS(PSI)
      ST=SIN(THA)
      CT=COS(THA)
92    TT=TAN(THA)
      R1=(V+EXDOT)*ST*CP+WYDOT*ST*SP+ZEDOT*CT
      R2=-(V+EXDOT)*SP+WYDOT*CP
      R3=(V+EXDOT)*CT*CP+WYDOT*CT*SP-ZEDOT*ST
      IF(R3.EQ.0) GO TO 6
      ARGA=R1/R3
      ARGB=R2/R3
      GO TO 11
6     IF(R1.EQ.0) GO TO 7
      GO TO 151
7     ARGA=0
      IF(R2.EQ.0) GO TO 12
      GO TO 151
12    ARGB=0
11    AHAT=ATAN(ARGA)
      BHAT=ATAN(ARGB)
      ALBAR=SQRT(AHAT**2+BHAT**2)
      SA=SIN(AHAT)
      CA=COS(AHAT)
      SB=SIN(BHAT)
      CB=COS(BHAT)
      SAL=SIN(ALBAR)
      CAL=COS(ALBAR)
      IF(ALBAR.EQ.0) GO TO 601
      CSFEHT=SA/SAL
      SNFEHT=SB/SAL
      GO TO 602
601   CSFEHT=1.0
      SNFEHT=0.0
602   FEEP=FEE+ARSIN(SNFEHT)
91    S4F=SIN(NF*FEEP)
      C4F=COS(NF*FEEP)
      SNFE=SIN(NF*FEE)
      CNFE=COS(NF*FEE)
      SNFAS=SIN(NF*ARSIN(SNFEHT))
      CNFAS=COS(NF*ARSIN(SNFEHT))
      CC1=C(1)*ALBAR+C(2)*ALBAR**3
      CC2=C(4)*ALBAR+C(5)*ALBAR**3
      CC3=C(6)*ALBAR+C(7)*ALBAR**3
      CC4=C(8)*ALBAR+C(9)*ALBAR**3
      CC5=C(10)+C(11)*ALBAR**2
      CC6=C(12)+C(13)*ALBAR**2
      CC7=C(14)*ALBAR+C(15)*ALBAR**3
      CC8=C(16)*ALBAR+C(17)*ALBAR**3
      CC9=C(18)*ALBAR+C(19)*ALBAR**3
      DC3=C(6)+3*C(7)*ALBAR**2
      DC1=C(1)+3*C(2)*ALBAR**2
      DC2=C(4)+3*C(5)*ALBAR**2
      DC4=C(8)+3*C(9)*ALBAR**2
      DC5=2*C(11)*ALBAR
      DC6=2*C(13)*ALBAR
      CC7=C(14)+3*C(15)*ALBAR**2
      DC8=C(16)+3*C(17)*ALBAR**2
      DC9=C(18)+3*C(19)*ALBAR**2
      D1=R1**2+R3**2
      D2=R2**2+R3**2
      DA1=(R3*ST*CP-R1*CT*CP)/D1
      DA2=(R3*ST*SP-R1*CT*SP)/D1
      DA3=(R3*CT+R1*ST)/D1
      DA4=(R3*R2*ST-R1*R3*CT)/D1
      DA5=1
      DB1=-(R3*SP+R2*CT*CP)/D2
      DB2=(R3*CP-R2*CT*SP)/D2
      DB3=(R2*ST)/D2
      DB4=-(R3*((V+EXDOT)*CP+WYDOT*SP)+R2**2*CT)/D2
      DB5=(R1*R2)/D2
      IF(ALBAR.EQ.0) GO TO 701
      E1=AHAT/ALBAR
      E2=BHAT/ALBAR
      E3=-(AHAT*SB*CAL)/(ALBAR*SAL**2)
      E4=CB/SAL-(BHAT*SB*CAL)/(ALBAR*SAL**2)
      E5=CA/SAL-(AHAT*SA*CAL)/(ALBAR*SAL**2)
      E6=-(BHAT*SA*CAL)/(ALBAR*SAL**2)
      GO TO 702
701   E1=0.0
```

```
      E2=0.0
      E3=0.0
      E4=0.0
      E5=0.0
      E6=0.0
  702 ED11=E1*DA4+E2*DB4
      ED12=E3*DA4+E4*DB4
      ED13=E5*DA4+E6*DB4
      ED21=E1*DA5+E2*DB5
      ED22=E3*DA5+E4*DB5
      ED23=E5*DA5+E6*DB5
      ED31=E1*DA1+E2*DB1
      ED32=E3*DA1+E4*DB1
      ED33=E5*DA1+E6*DB1
      ED41=E1*DA2+E2*DB2
      ED42=E3*DA2+E4*DB2
      ED43=E5*DA2+E6*DB2
      ED51=E1*DA3+E2*DB3
      ED52=E3*DA3+E4*DB3
      ED53=E5*DA3+E6*DB3
      DO 1000 I=1,NEQ
      KT=IEQ(I)
      GO TO (21,22,23,24,25,26),KT
C     PSI EQUATION HOMOGENEOUS TERMSO
   21 F11=-DC5*PSIDOT-DC4*FEEDOT*CSFEHT/CT+DC2*SNFEHT/CT-DC3
     1*S4F*CSFEHT/CT-CC3*CSFEHT/CT*NF*SB*CAL*
     2(SNFE*SNFAS+CNFE*CNFAS)/(SAL*SQRT(SAL**2-SB**2))
      F112=CC3*CSFEHT/CT*NF*CB*(SNFE*SNFAS+CNFE*CNFAS)/
     1SQRT(SAL**2-SB**2)
      F12=CC2/CT
      F13=-CC4*FEEDOT/CT-CC3*S4F/CT
      FSX(2,1)=F11*ED11+F12*ED12+F13*ED13+F112*E2*DB4
      FSX(2,2)=-(2*THADOT*TT+CC5)
      FSX(2,3)=-ST/CT**2*(2*PSIDOT*THADOT*ST+AK1*THADOT+
     1CC4*FEEDOT*CSFEHT-CC2*SNFEHT+CC3*S4F*CSFEHT)+F11*ED21+F12*ED22
     2+F13*ED23+F112*E2*DB5
      FSX(2,4)=-2*PSIDOT*TT-AK1/CT
      FSX(2,5)=-NF*CC3*CSFEHT/CT*(CNFAS*CNFE-SNFAS*SNFE)
      FSX(2,6)=-CC4*CSFEHT/CT
      FSX(2,8)=F11*ED31+F12*ED32+F13*ED33+F112*E2*DB1
      FSX(2,10)=F11*ED41+F12*ED42+F13*ED43+F112*E2*DB2
      FSX(2,12)=F11*ED51+F12*ED52+F13*ED53+F112*E2*DB3
      GO TO 1000
C     THETA EQUATION HOMOGENEOUS TERMSO
22    F21=-DC5*THADOT-CSFEHT*DC2-FEEDOT*SNFEHT*DC4-S4F*
     1SNFEHT*DC3-CC3*SNFEHT*NF*SB*CAL*
     2(SNFE*SNFAS+CNFE*CNFAS)/(SAL*SQRT(SAL**2-SB**2))
      F22=-CC4*FEEDOT-CC3*S4F
      F212=CC3*SNFEHT*NF*CB*(SNFAS*SNFE+CNFAS*CNFE)/
     1SQRT(SAL**2-SB**2)
      F23=-CC2
      FSX(4,1)=F21*ED11+F23*ED13+F22*ED12+F212*E2*DB4
      FSX(4,2)=AK1*CT+PSIDOT*2*ST*CT
      FSX(4,3)=PSIDOT**2*(CT**2-ST**2)-PSIDOT*AK1*ST+F23*ED23
     1+F21*ED21+F22*ED22+F212*E2*DB5
      FSX(4,4)=-CC5
      FSX(4,5)=-NF*CC3*SNFEHT*(CNFAS*CNFE-SNFAS*SNFE)
      FSX(4,6)=-CC4*SNFEHT
      FSX(4,8)=F21*ED31+F22*ED32+F23*ED33+F212*E2*DB1
      FSX(4,10)=F21*ED41+F22*ED42+F23*ED43+F212*E2*DB2
      FSX(4,12)=F21*ED51+F22*ED52+F23*ED53+F212*E2*DB3
      GO TO 1000
C     FEE EQUATION HOMOGENEOUS TERMSO
23    IF(IEQ(1).EQ.1.AND.IEQ(2).EQ.0) GO TO 231
      IF(ALBAR.EQ.0) GO TO 231
      F31=-S4F*DC1-NF*SB*CAL*(SNFE*SNFAS+CNFE*CNFAS)/(SAL*
     1SQRT(SAL**2-SB**2))*CC1
      F312=+NF*CB/SQRT(SAL**2-SB**2)*(SNFE*SNFAS+CNFAS*CNFE)*CC1
      GO TO 232
231   F31=-S4F*DC1
      F312=0.0
232   FSX(6,1)=F31*ED11+F312*E2*DB4
      FSX(6,2)=THADOT*CT
      FSX(6,3)=F31*ED21-X(1)*THADOT*CT+PSIDOT*THADOT**2*ST
      FSX(6,4)=-PSIDOT*CT
      FSX(6,5)=-NF*(CNFAS*CNFE-SNFAS*SNFE)*CC1
      FSX(6,6)=-C(3)
      FSX(6,8)=F31*ED31+F312*E2*DB1
      FSX(6,10)=F31*ED41+F312*E2*DB2
      FSX(6,12)=F31*ED51+F312*E2*DB3
      GO TO 1000
C     EX EQUATION HOMOGENEOUS TERMSO
24    AT1=SP*CSFEHT+ST*CP*SNFEHT
      AT2=SP*SNFEHT-ST*CP*CSFEHT
      F41=FEEDOT*AT1*DC8-DC6*CT*CP+S4F*AT1*DC7+AT2*DC9
     1+AT1*CC7*NF*SB*CAL*(SNFE*SNFAS+CNFE*CNFAS)/
     2(SAL*SQRT(SAL**2-SB**2))
      F412=-AT1*CC7*NF*CB*(SNFE*SNFAS+CNFE*CNFAS)/
     1SQRT(SAL**2-SB**2)
      F42=CC8*FEEDOT*ST*CP+CC7*S4F*ST*CP+CC9*SP
      F43=CC8*FEEDOT*SP+CC7*S4F*SP-CC9*ST*CP
      FSX(8,1)=CP*(CC8*FEEDOT*CSFEHT+CC7*S4F*CSFEHT+CC9*SNFEHT)-
     1SP*(CC8*FEEDOT*ST*SNFEHT+CC7*S4F*ST*SNFEHT-CC6*CT-CC9*ST*
     2CSFEHT)+F41*ED11+F42*ED12+F43*ED13+F412*E2*DB4
      FSX(8,3)=CT*(CC8*FEEDOT*CP*SNFEHT+CC7*S4F*CP*SNFEHT-CC9*
     1CP*CSFEHT)+CC6*ST*CSFEHT+F41*ED21+F42*ED22+F43*ED23
     2+F412*E2*DB5
      FSX(8,5)=AT1*CC7*NF*(CNFAS*CNFE-SNFAS*SNFE)
      FSX(8,6)=CC8*AT1
      FSX(8,8)=F41*ED31+F42*ED32+F43*ED33+F412*E2*DB1
      FSX(8,10)=F41*ED41+F42*ED42+F43*ED43+F412*E2*DB2
      FSX(8,12)=F41*ED51+F42*ED52+F43*ED53+F412*E2*DB3
      GO TO 1000
C     WY EQUATION HOMOGENEOUS TERMSO
25    AT3=-CP*CSFEHT+ST*SP*SNFEHT
      AT4=CP*SNFEHT+ST*SP*CSFEHT
      F51=AT3*(FEEDOT*DC8+S4F*DC7)-AT4*DC9-DC6*CT*SP
     1+AT3*CC7*NF*SB*CAL*(SNFE*SNFAS+CNFE*CNFAS)/
     2(SAL*SQRT(SAL**2-SB**2))
      F512=-AT3*CC7*NF*CB*(SNFE*SNFAS+CNFE*CNFAS)/
     1SQRT(SAL**2-SB**2)
      F52=CC8*FEEDOT*ST*SP+CC7*S4F*ST*SP-CC9*CP
      F53=-(CC8*FEEDOT*CP+CC7*S4F*CP+CC9*ST*SP)
      FSX(10,1)=CP*(CC8*FEEDOT*ST*SNFEHT-CC6*CT+CC7*S4F*ST*SNFEHT
     1-CC9*ST*CSFEHT)+SP*(CC8*FEEDOT*CSFEHT+CC7*S4F*CSFEHT+
     2CC9*SNFEHT)+F51*ED11+F52*ED12+F53*ED13+F512*E2*DB4
      FSX(10,3)=CT*(CC8*FEEDOT*SP*SNFEHT+CC7*S4F*SP*SNFEHT-CC9*
     1SP*CSFEHT)+ST*(CC6*SP)+F51*ED21+F52*ED22+F53*ED23
     2+F512*E2*DB5
      FSX(10,5)=-AT3*CC7*NF*(CNFAS*CNFE-SNFAS*SNFE)
      FSX(10,6)=CC8*AT3
      FSX(10,8)=F51*ED31+F52*ED32+F53*ED33+F512*E2*DB1
      FSX(10,10)=F51*ED41+F52*ED42+F53*ED43+F512*E2*DB2
      FSX(10,12)=F51*ED51+F52*ED52+F53*ED53+F512*E2*DB3
      GO TO 1000
C     ZE EQUATION HOMOGENEOUS TERMSO
26    AT5=CT*SNFEHT
      AT6=CT*CSFEHT
      F61=AT5*(FEEDOT*DC8+S4F*DC7)-AT6*DC9+DC6*ST
     1+AT5*CC7*NF*SB*CAL*(SNFE*SNFAS+CNFE*CNFAS)/
     2(SAL*SQRT(SAL**2-SB**2))
      F612=-AT5*CC7*NF*CB*(SNFE*SNFAS+CNFE*CNFAS)/
     1SQRT(SAL**2-SB**2)
      F62=CC8*FEEDOT*CT-CC7*S4F*CT
      F63=-CC9*CT
      FSX(12,1)=F61*ED11+F62*ED12+F63*ED13+F612*E2*DB4
      FSX(12,3)=CC6*CT-ST*(CC8*FEEDOT*SNFEHT+CC7*S4F*SNFEHT-
     1CC9*CSFEHT)+F61*ED21+F62*ED22+F63*ED23+F612*E2*DB5
      FSX(12,5)=-AT5*CC7*NF*(CNFE*CNFAS-SNFE*SNFAS)
      FSX(12,6)=CC8*CT*SNFEHT
      FSX(12,8)=F61*ED31+F62*ED32+F63*ED33+F612*E2*DB1
      FSX(12,10)=F61*ED41+F62*ED42+F63*ED43+F612*E2*DB2
      FSX(12,12)=F61*ED51+F62*ED52+F63*ED53+F612*E2*DB3
 1000 CONTINUE
 10   DO 100 M=1,NP
      DO 9991 JJ=1,12
      J=JJ+(M-1)*12
      DOT(J)=0.0
      GO TO (98,99,98,99,98,99,98,99,98,99,98,99),JJ
 98   K=JJ+1
      L=J+1
      DOT(J)=X(L)
      GO TO 9991
 99   DO 991 K=1,12
      L=K+(M-1)*12
 991  DOT(J)=DOT(J)-FSX(JJ,K)*X(L)
 9991 CONTINUE
 100  CONTINUE
      II=NIC*12-12
      IF(NC)19,27,19
 19   DO 20 M=1,NC
      JC=CADJ(M)
      JX=II+M*12
      GO TO (101,102,103,104,105,106,107,108,109,110,111,112,113,
     1114,115,116,117,118,119),JC
 101  DOT(JX+6)=DOT(JX+6)+ALBAR*S4F
      GO TO 20
 102  DOT(JX+6)=DOT(JX+6)+ALBAR**3*S4F
      GO TO 20
 103  DOT(JX+6)=DOT(JX+6)+FEEDOT
```

```
      GO TO 20
104   DOT(JX+2)=DOT(JX+2)-ALBAR*SNFEHT/CT
      DOT(JX+4)=DOT(JX+4)+ALBAR*CSFEHT
      GO TO 20
105   DOT(JX+2)=DOT(JX+2)-ALBAR**3*SNFEHT/CT
      DOT(JX+4)=DOT(JX+4)+ALBAR**3*CSFEHT
      GO TO 20
106   DOT(JX+2)=DOT(JX+2)+ALBAR*S4F*CSFEHT/CT
      DOT(JX+4)=DOT(JX+4)+ALBAR*S4F*SNFEHT
      GO TO 20
107   DOT(JX+2)=DOT(JX+2)+ALBAR**3*S4F*CSFEHT/CT
      DOT(JX+4)=DOT(JX+4)+ALBAR**3*S4F*SNFEHT
      GO TO 20
108   DOT(JX+2)=DOT(JX+2)+ALBAR*FEEDOT*CSFEHT/CT
      DOT(JX+4)=DOT(JX+4)+ALBAR*FEEDOT*SNFEHT
      GO TO 20
109   DOT(JX+2)=DOT(JX+2)+ALBAR**3*FEEDOT*CSFEHT/CT
      DOT(JX+4)=DOT(JX+4)+ALBAR**3*FEEDOT*SNFEHT
      GO TO 20
110   DOT(JX+2)=DOT(JX+2)+PSIDOT
      DOT(JX+4)=DOT(JX+4)+THADOT
      GO TO 20
111   DOT(JX+2)=DOT(JX+2)+ALBAR**2*PSIDOT
      DOT(JX+4)=DOT(JX+4)+ALBAR**2*THADOT
      GO TO 20
112   DOT(JX+8)=DOT(JX+8)+CT*CP
      DOT(JX+10)=DOT(JX+10)+CT*SP
      DOT(JX+12)=DOT(JX+12)-ST
      GO TO 20
113   DOT(JX+8)=DOT(JX+8)+ALBAR**2*CT*CP
      DOT(JX+10)=DOT(JX+10)+ALBAR**2*CT*SP
      DOT(JX+12)=DOT(JX+12)-ALBAR**2*ST
      GO TO 20
114   DOT(JX+8)=DOT(JX+8)-ALBAR*S4F*AT1
```

```
      DOT(JX+10)=DOT(JX+10)-ALBAR*S4F*AT3
      DOT(JX+12)=DOT(JX+12)-ALBAR*S4F*AT5
      GO TO 20
115   DOT(JX+8)=DOT(JX+8)-ALBAR**3*S4F*AT1
      DOT(JX+10)=DOT(JX+10)-ALBAR**3*S4F*AT3
      DOT(JX+12)=DOT(JX+12)-ALBAR**3*S4F*AT5
      GO TO 20
116   DOT(JX+8)=DOT(JX+8)-ALBAR*FEEDOT*AT1
      DOT(JX+10)=DOT(JX+10)-ALBAR*FEEDOT*AT3
      DOT(JX+12)=DOT(JX+12)-ALBAR*FEEDOT*AT5
      GO TO 20
117   DOT(JX+8)=DOT(JX+8)-ALBAR**3*FEEDOT*AT1
      DOT(JX+10)=DOT(JX+10)-ALBAR**3*FEEDOT*AT3
      DOT(JX+12)=DOT(JX+12)-ALBAR**3*FEEDOT*AT5
      GO TO 20
118   DOT(JX+6)=DOT(JX+8)-ALBAR*AT2
      DOT(JX+10)=DOT(JX+10)+ALBAR*AT4
      DOT(JX+12)=DOT(JX+12)+ALBAR*AT6
      GO TO 20
119   DOT(JX+8)=DOT(JX+8)-ALBAR**3*AT2
      DOT(JX+10)=DOT(JX+10)+ALBAR**3*AT4
      DOT(JX+12)=DOT(JX+12)+ALBAR**3*AT6
20    CONTINUE
27    CONTINUE
152   RETURN
151   RETURN 1
      END
      SUBROUTINE OUT2(KDUM)
      INTEGER CADJ(19)
      COMMON N,T,X(2232)
      COMMON V,RO,AR,C,AIX,AI,AM,G,P
      COMMON DATUM(12,500),DCALC(12,500)
      COMMON /DATA1/NEQ,NIC,NC,NP,CADJ,QW(12),AK1,NPTS,KKK,
     1C(19),H,B(31),IEQ(6),MODE,JJJ,NF,ICADJ(12),JT
      COMMON /DATA2/AJK(31,31)
   27 DO 150 M=1,NP
      II=(M-1)*12
      DO 149 K=1,12
      IF(QW(K).EQ.0) GO TO 149
      B(M)=B(M)+X(II+K)*QW(K)*(DATUM(K,JT)-DCALC(K,JT))
      DO 148 J=1,M
      JJ=(J-1)*12
      AJK(M,J)=AJK(M,J)+X(II+K)*X(JJ+K)*QW(K)
  147 IF(M.EQ.J) GO TO 148
      AJK(J,M)=AJK(M,J)
  148 CONTINUE
  149 CONTINUE
  150 CONTINUE
      RETURN
      END
      SUBROUTINE MINV(A,N,D,L,M)                                        MINV 330
      DIMENSION A(1),L(1),M(1)                                          MINV 340
      D=1.0                                                             MINV 560
      NK=-N                                                             MINV 570
      DO 80 K=1,N                                                       MINV 580
      NK=NK+N                                                           MINV 590
      L(K)=K                                                            MINV 600
      M(K)=K                                                            MINV 610
      KK=NK+K                                                           MINV 620
      BIGA=A(KK)                                                        MINV 630
      DO 20 J=K,N                                                       MINV 640
      IZ=N*(J-1)                                                        MINV 650
      DO 20 I=K,N                                                       MINV 660
      IJ=IZ+I                                                           MINV 670
   10 IF( ABS(BIGA)- ABS(A(IJ))) 15,20,20                               MINV 680
   15 BIGA=A(IJ)                                                        MINV 690
      L(K)=I                                                            MINV 700
      M(K)=J                                                            MINV 710
   20 CONTINUE                                                          MINV 720
      J=L(K)                                                            MINV 760
      IF(J-K) 35,35,25                                                  MINV 770
   25 KI=K-N                                                            MINV 780
      DO 30 I=1,N                                                       MINV 790
      KI=KI+N                                                           MINV 800
      HOLD=-A(KI)                                                       MINV 810
      JI=KI-K+J                                                         MINV 820
      A(KI)=A(JI)                                                       MINV 830
   30 A(JI) =HOLD                                                       MINV 840
   35 I=M(K)                                                            MINV 880
      IF(I-K) 45,45,38                                                  MINV 890
   38 JP=N*(I-1)                                                        MINV 900
      DO 40 J=1,N                                                       MINV 910
      JK=NK+J                                                           MINV 920
      JI=JP+J                                                           MINV 930
      HOLD=-A(JK)                                                       MINV 940
      A(JK)=A(JI)                                                       MINV 950
   40 A(JI) =HOLD                                                       MINV 960
   45 IF(BIGA) 48,46,48                                                 MINV1010
   46 D=0.0                                                             MINV1020
      RETURN                                                            MINV1030
   48 DO 55 I=1,N                                                       MINV1040
      IF(I-K) 50,55,50                                                  MINV1050
   50 IK=NK+I                                                           MINV1060
      A(IK)=A(IK)/(-BIGA)                                               MINV1070
   55 CONTINUE                                                          MINV1080
      DO 65 I=1,N                                                       MINV1120
      IK=NK+I                                                           MINV1130
      HOLD=A(IK)                                                        MINV1140
      IJ=I-N                                                            MINV1150
      DO 65 J=1,N                                                       MINV1160
      IJ=IJ+N                                                           MINV1170
      IF(I-K) 60,65,60                                                  MINV1180
   60 IF(J-K) 62,65,62                                                  MINV1190
   62 KJ=IJ-I+K                                                         MINV1200
      A(IJ)=HOLD*A(KJ)+A(IJ)                                            MINV1210
   65 CONTINUE                                                          MINV1220
      KJ=K-N                                                            MINV1260
      DO 75 J=1,N                                                       MINV1270
      KJ=KJ+N                                                           MINV1280
      IF(J-K) 70,75,70                                                  MINV1290
   70 A(KJ)=A(KJ)/BIGA                                                  MINV1300
   75 CONTINUE                                                          MINV1310
      D=D*BIGA                                                          MINV1350
      A(KK)=1.0/BIGA                                                    MINV1390
   80 CONTINUE                                                          MINV1400
      K=N                                                               MINV1440
  100 K=(K-1)                                                           MINV1450
      IF(K) 150,150,105                                                 MINV1460
```

```
105 I=L(K)                                                              MINV1470
    IF(I-K) 120,120,108                                                 MINV1480
108 JQ=N*(K-1)                                                          MINV1490
    JR=N*(I-1)                                                          MINV1500
    DO 110 J=1,N                                                        MINV1510
    JK=JQ+J                                                             MINV1520
    HOLD=A(JK)                                                          MINV1530
    JI=JR+J                                                             MINV1540
    A(JK)=-A(JI)                                                        MINV1550
110 A(JI) =HOLD                                                         MINV1560
120 J=M(K)                                                              MINV1570
    IF(J-K) 100,100,125                                                 MINV1580
125 KI=K-N                                                              MINV1590
    DO 130 I=1,N                                                        MINV1600
    KI=KI+N                                                             MINV1610
    HOLD=A(KI)                                                          MINV1620
    JI=KI-K+J                                                           MINV1630
    A(KI)=-A(JI)                                                        MINV1640
130 A(JI) =HOLD                                                         MINV1650
    GO TO 100                                                           MINV1660
150 RETURN                                                              MINV1670
    END                                                                 MINV1680
```

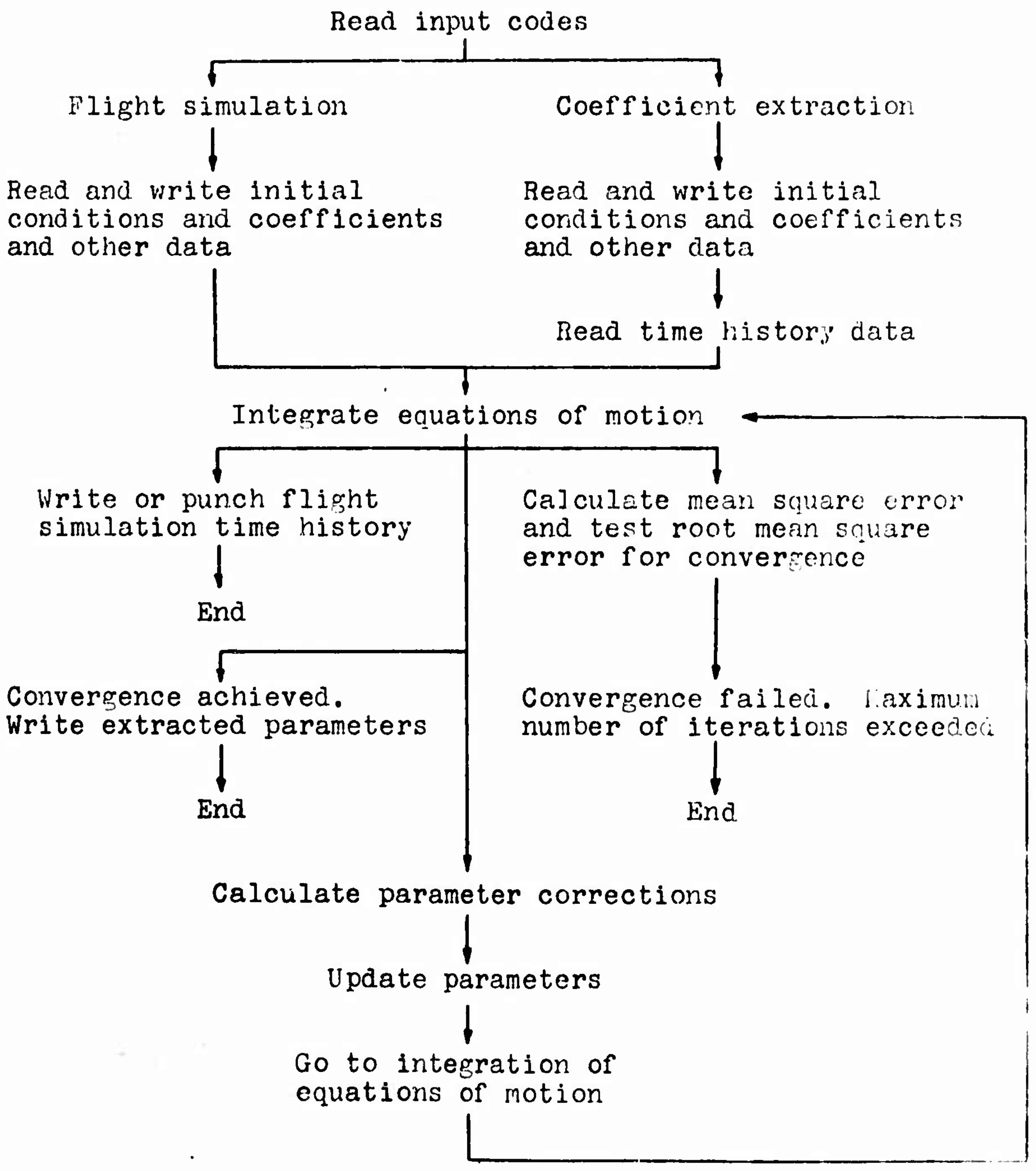
```
      SUBROUTINE PLOT9(Y,NK,NPL)
      REAL*4 LINE1(121),LINE2(121),Y(NK,NPL),FMT(6),SYM(9),FORMS(3)
      INTEGER*4 J1(9)
      REAL*8 SYMB(9)
      DATA FMT/'(2X,','121 ','A1,3','X, 1','9A10',')    '/,
     1FIVA10/4H9A10/,SIX/4H)    /
      DATA SYMB/' Y(X,1) ',' Y(X,2) ',' Y(X,3) ',' Y(X,4) ',' Y(X,5) ',
     1' Y(X,6) ',' Y(X,7) ',' Y(X,8) ',' Y(X,9) '/
      DATA STAR,X,BLANK,PLUS,ZERO/4H****,4HXXXX,4H    ,4H++++,4HIIII/
      DATA FORMS/'E   ','F   ','X   '/
      DATA SYM/4HXXXX,4H0000,4H0000,4H4444,4H5555,4H6666,4H7777,4H8888,
     14H9999/,FIVE/4HP9E1/,SIXE/4H0.2)/,FIVEF/4H9F7./,SIXF/4H2)  /,
     2FIVA7/4H9A7 /,INT/0/
      FMT(5)=FIVA10
      FMT(6)=SIX
      READ (5,2) FMT(2),LMAX,LO,LOG,FORM,SF
    2 FORMAT  (A3,T1,I3,T5,I3,T9,I1,T11,A1,T13,F10.3)
      IF(LO.LT.1)    LO=1
      IF(LO.GT.LMAX) LO=LMAX
      DO 10 I=1,LMAX
      LINE1(I)=PLUS
   10 LINE2(I)=BLANK
      DO 20 I=1,LMAX,10
      IF((LO+1-I).GT.0)    L1=LO+1-I
      IF((LO-1+I).LT.(LMAX+1))  L2=LO-1+I
   20 CONTINUE
      DO 30 I=L1,L2,10
   30 LINE2(I)=PLUS
      LINE1(LO)=ZERO
      LINE2(LO)=ZERO
      IF (LOG .GT. 0) GO TO 51
      GO TO 42
   51 IS=LOG*10
      WRITE (6,300) IS,SF
```

```
300     FORMAT (14X,I2,'*LOG10(',1PE10.3,' * Y )'//)
42      IP=NPL
        IF (FORM .EQ. FORMS(3)) GO TO 52
        IF (FORM .EQ. FORMS(2)) GO TO 43
        IF ((LMAX + 10*NPL) .GT. 135) IP=(135-LMAX)/10
        GO TO 53
43      IF ((LMAX + 7*NPL) .GT. 135) IP=(135-LMAX)/7
        FMT(5)=FIVA7
53      IF (IP .LT. 1) GO TO 52
        GO TO 44
52      WRITE(6,FMT) (LINE1(M),M=1,LMAX)
        GO TO 45
44      WRITE (6,FMT) (LINE1(M),M=1,LMAX),(SYMB(M),M=1,IP)
45      FMT(5)=FIVE
        FMT(6)=SIXE
        IF (FORM .EQ. FORMS(3) ) IP = 0
        IF(FORM .EQ. FORMS(2) ) GO TO 16
        GO TO 46
16      FMT(5)=FIVEF
        FMT(6)=SIXF
46      K=10
        DO 60 I=1,NK
        DO 11 I2=1,NPL
        IF (LOG .GT.0) GO TO 47
        J1(I2)=Y(I,I2) * SF + LO + 0.5
        GO TO 48
47      J1(I2)=IS * ALOG10(ABS(Y(I,I2) * SF)) + LO + 0.5
48      IF(J1(I2) .GT. LMAX) J1(I2)=LMAX
        IF(J1(I2) .LT. 1 )   J1(I2)=1
11      CONTINUE
        IF (I-K) 40,70,40
70      DO 12  I2=1,NPL
        J=J1(I2)
12      LINE1(J)=SYM(I2)
```

```
        IF (IP .LT. 1) GO TO 61
        Y(I,M)=Y(I,M)-0.6
        WRITE (6,FMT) (LINE1(M), M=1,LMAX),(Y(I,M), M=1,IP)
        GO TO 62
61      WRITE (6,FMT) (LINE1(M), M=1,LMAX)
62      CONTINUE
        DO 13 I2=1,NPL
        J=J1(I2)
13      LINE1(J)=PLUS
        LINE1(LO)=ZERO
        K=K + 10
        GO TO 60
40      DO 14  I2=1,NPL
        J=J1(I2)
14      LINE2(J)=SYM(I2)
        IF(IP .LT. 1) GO TO 63
        Y(I,M)=Y(I,M)-0.6
        WRITE (6,FMT) (LINE2(M), M=1,LMAX),(Y(I,M), M=1,IP)
        GO TO 64
63      WRITE (6,FMT) (LINE2(M), M=1,LMAX)
64      CONTINUE
        DO 15 I2=1,NPL
        J=J1(I2)
15      LINE2(J)=BLANK
        DO 500 I2=L1,L2,10
500      LINE2(I2)=PLUS
        LINE2(LO)=ZERO
60      CONTINUE
        WRITE (6,111)
111     FORMAT(//)
        INT=1
        RETURN
        END
/*
```

APPENDIX II

FLOW CHART OF COMPUTER PROGRAM


Read input codes
Flight simulation
Coefficient extraction
Read and write initial conditions and coefficients and other data
Read and write initial conditions and coefficients and other data
Read time history data
Integrate equations of motion
Write or punch flight simulation time history
Calculate mean square error and test root mean square error for convergence
End
Convergence achieved. Write extracted parameters
Convergence failed. Maximum number of iterations exceeded
End
End
Calculate parameter corrections
Update parameters
Go to integration of equations of motion

APPENDIX III

DATA INPUT FORMAT

CASE I Flight simulation

CARD	COLUMNS	FIELD	EXPLANATION
1	1	I1	Mode = 0
2	1-6	I1	IEQ(I) Equations of motion to be integrated. In order, the equations are:
			Yawing (ψ)
			Pitching (θ)
			Rolling (ϕ)
			X translation (X)
			Y translation (Y)
			Z translation (Z)
3			Integration constants
	1-5	F5.3	H
	6-8	I3	ITO
	9-13	F5.3	TMAX
	14-18	F5.3	TZERO
	19	I1	IP 0:no, 1:yes
	20	I1	IT 0:no, 1:yes
	21	I1	IFE 0:no, 1:yes
	22	.I1	NF
4			Aerodynamic constants
	1-11	F11.8	RO
	12-22	F11.4	V

CARD	COLUMNS	FIELD	EXPLANATION
	23-33	F11.4	AR
	34-44	F11.4	D
	45-55	F11.4	P
5			Aerodynamic constants
	1-11	F11.8	g
	12-22	F11.4	AIX
	13-33	F11.4	AI
	34-44	F11.4	AM
6			Aerodynamic coefficient values
	1-10	F10.4	CLA
	11-20	F10.4	CLA3
	21-30	F10.4	CLP
	31-40	F10.4	CMA
	41-50	F10.4	CMA3
	51-60	F10.4	CNA
	61-70	F10.4	CNA3
	71-80	F10.4	CNPA
7			Aerodynamic coefficient values
	1-10	F10.4	CNPA3
	11-20	F10.4	CMQO
	21-30	F10.4	CMQ2
	31-40	F10.4	CXO
	41-50	F10.4	CXA2
	51-60	F10.4	CYA
	61-70	F10.4	CYA3

CARD	COLUMNS	FIELD	EXPLANATION
	71-80	F10.4	CYPA
8			Aerodynamic coefficient values
	1-10	F10.4	CYPA3
	11-20	F10.4	CZA
	21-30	F10.4	CZA3
9			Initial condition values
	1-10	F10.4	ψ_o
	11-20	F10.4	$\dot{\psi}_o$
	21-30	F10.4	θ_o
	31-40	F10.4	$\dot{\theta}_o$
	41-50	F10.4	ϕ_o
	51-60	F10.4	$\dot{\phi}_o$
	61-70	F10.4	X_o
	71-80	F10.4	$\dot{X}_o$
10			Initial condition values
	1-10	F10.4	Y_o
	11-20	F10.4	$\dot{Y}_o$
	21-30	F10.4	Z_o
	31-40	F10.4	$\dot{Z}_o$
	41-50	F10.4	MSQE (always 0.0)
11 (12,13)			PLOT9 data (one card for each plot desired of ψ, θ, or ϕ)

CARD	COLUMNS	FIELD	EXPLANATION
	1-3	A3	FMT(2) Width of format for output (same as width of plot)
	1-3	I3	LMAX: Width of plot (1≤LMAX≤121)
	5-7	I3	LO: Value of initial point
	9	I1	LOG: Type of plot
			0 - Linear
			1 - Log
			2 - Log-log
	11	A1	FORM: Form of data point printed on plot
			F - F field
			E - E field
			X - No data point value printed
	13-22	F10.3	Scale factor

Case II Coefficient extraction

CARD	COLUMNS	FIELD	EXPLANATION
1	1	I1	Mode = 1
2-5			Labels for output (listed at end of appendix)
6	1-12	I1	ICADJ(I) Initial conditions to be adjusted 1:yes, 0:no In order they are: ψ_o, $\dot{\psi}_o$, θ_o, $\dot{\theta}_o$, ϕ_o, $\dot{\phi}_o$, X_o, $\dot{X}_o$, Y_o, $\dot{Y}_o$, Z_o, $\dot{Z}_o$
7	1-19	I1	CADJ(I) Coefficients to be adjusted 0:no, 1:yes In order they are: $C_{\ell\bar{\alpha}}$, $C_{\ell\bar{\alpha}^3}$, $C_{\ell p}$, $C_{m\bar{\alpha}}$, $C_{m\bar{\alpha}^3}$, $C_{n\bar{\alpha}}$, $C_{n\bar{\alpha}^3}$, $C_{np\bar{\alpha}}$ $C_{np\bar{\alpha}^3}$, C_{mq_o}, C_{mq2}, C_{x_o}, $C_{x\bar{\alpha}^2}$, $C_{y\bar{\alpha}}$ $C_{y\bar{\alpha}^3}$ $C_{yp\bar{\alpha}}$, $C_{yp\bar{\alpha}}$, $C_{z\alpha}$ $C_{z\bar{\alpha}^3}$
8	1-60	F5.2	QW(I) Weight factors for state vector components (same order as initial conditions)
9			Convergence criteria
	1-2	I2	MAXIT Maximum number of iterations allowed before program is terminated
	3-12	F10.4	TOL Convergence tolerance
10	1-6	I1	IEQ(I) 0:no, 1:yes
11	1-12	I1	ISV(I) State vector component time histories to be read as data 0:no 1:yes (same order as initial conditions)
12			Integration constants
	1-5	F5.3	H
	6-8	I3	ITO
	9-13	F5.3	TMAX
	14-18	F5.3	TZERO

CARD	COLUMNS	FIELD	EXPLANATION
	19	I1	IP 0:no 1:yes
	20	I1	IT 0:no 1:yes
	21	I1	IFE 0:no 1:yes
	22	I1	NF
13			Aerodynamic constants
	1-11	F11.8	RO
	12-22	F11.4	V
	23-33	F11.4	AR
	34-44	F11.4	D
	45-55	F11.4	P
14			Aerodynamic constants
	1-11	F11.8	g
	12-22	F11.4	AIX
	23-33	F11.4	AI
	34-44	F11.4	AM
15			Aerodynamic coefficient estimates
	1-10	F10.4	CLA
	11-20	F10.4	CLA3
	21-30	F10.4	CLP
	31-40	F10.4	CMA
	41-50	F10.4	CMA3
	51-60	F10.4	CNA
	61-70	F10.4	CNA3
	71-80	F10.4	CNPA

CARD	COLUMNS	FIELD	EXPLANATION
16			Aerodynamic coefficient estimates
	1-10	F10.4	CNPA3
	11-20	F10.4	CMQO
	21-30	F10.4	CMQ2
	31-40	F10.4	CXO
	41-50	F10.4	CXA2
	51-60	F10.4	CYA
	61-70	F10.4	CYA3
	71-80	F10.4	CYPA
17			Aerodynamic coefficient estimates
	1-10	F10.4	CYPA3
	11-20	F10.4	CZA
	21-30	F10.4	CZA3
NCARDS	1-80	5E15.6	Time histories of state vector components NCARDS = NSV(NPTS/5)
18			Initial condition estimates
NCARDS			
	1-10	F10.4	ψ_o
	11-20	F10.4	$\dot{\psi}_o$
	21-30	F10.4	θ_o
	31-40	F10.4	$\dot{\theta}_o$
	41-50	F10.4	ϕ_o
	51-60	F10.4	$\dot{\phi}_o$

CARD	COLUMNS	FIELD	EXPLANATION
	61-70	F10.4	X_o
	71-80	F10.4	$\dot{X}_o$
19			Initial condition estimates
NCARDS			
	1-10	F10.4	Y_o
	11-20	F10.4	$\dot{Y}_o$
	21-30	F10.4	Z_o
	31-40	F10.4	$\dot{Z}_o$
	41-50	F10.4	MSQE (always 0.0)
20,21,22			Same as PLOT9 data in Case I
NCARDS			

Case III Flight simulation with punched output

Same as Case I except a card like card #11 of Case II is inserted between cards #2 and #3 of Case I.

Output Labels (Cards 2-5) For Case II Coefficient Extraction

```
Column   1         11          21        31         41         51         61         71
         ↓         ↓           ↓         ↓          ↓          ↓          ↓          ↓
Card #2  SCLASCL3SCLPSCMASCM3SCNASCN3SCNPSNP3SMQOSMQ2SCXOSCX2SCYASCY3SCYPSYP3SCZASCZ3
Card #3   CLACLA3 CLP CMACMA3 CNACNA3CNPACNP3OMQOCMQ2 CXOCXA2 CYACYA3CYAPCYP3 CZACZA3
Card #4   SPOSPDO STOSTDO SFOSFDO SXOSXDO SYOSYDO SZOSZDO
Card #5  PSIOPDTOTHAOTDTOFEEOFDTO  XOXDTO  YOYDTO  ZOZDTO
```

APPENDIX IV

RESULTS OF PROGRAM TEST RUNS

RUN	DEGREES OF FREEDOM	PARAMETER EXTRACTED	INITIAL GUESS	EXTRACTED VALUE	ESTIMATED STANDARD DEVIATION	CORRECT VALUE
1	1-Noise	θ_o	0.1800	0.1754	0.00129	0.1754*
	Linear	$\dot{\theta}_o$	0.0	0.0106	0.0184	0.0106
		$C_{m\bar{\alpha}}$	-1.750	-2.011	0.00541	-2.010
		C_{mqo}	-65.000	-60.669	1.1834	-60.585
2	1-No Noise	θ_o	0.1800	0.1745	0.000003	0.1745
	Linear	$\dot{\theta}_o$	0.0	0.0000	0.00004	0.0
		$C_{m\bar{\alpha}}$	-1.750	-2.002	0.00001	-2.000
		C_{mqo}	-65.000	-60.108	0.0026	-60.000
3	1-Noise	θ_o	0.5235	0.5260	0.00144	0.5259*
	Nonlinear	$\dot{\theta}_o$	0.0	0.0412	0.07249	0.0411
		$C_{m\bar{\alpha}}$	-1.800	-2.013	0.00799	-2.012
		$C_{m\bar{\alpha}^3}$	-25.000	-24.613	0.14766	-24.600
		C_{mqo}	-62.000	-61.660	0.73425	-61.578
		C_{mq2}	-175.000	-239.086	23.947	-238.630
4	1-No Noise	ψ_o	0.5235	0.5235	0.00003	0.5235
	Nonlinear	$\dot{\psi}_o$	0.0	0.0000	0.00152	0.0
		$C_{m\bar{\alpha}}$	-1.800	-2.000	0.00018	-2.000
		$C_{m\bar{\alpha}^3}$	-25.000	-24.500	0.00316	-24.500
		C_{mqo}	-62.000	-60.000	0.01551	-60.000
		C_{mq2}	-175.000	-160.004	0.48508	-160.000

*Values obtained from UFPLANAR

RUN	DEGREES OF FREEDOM	PARAMETER EXTRACTED	INITIAL GUESS	EXTRACTED VALUE	ESTIMATED STANDARD DEVIATION	CORRECT VALUE
5	1-No Noise	θ_o	0.5235	0.5235	0.00003	0.5235
	Nonlinear	$\dot{\theta}_o$	0.0	0.0000	0.00152	0.0
		$C_{m\bar{\alpha}}$	-1.800	-2.000	0.00018	-2.000
		$C_{m\bar{\alpha}^3}$	-25.000	-24.500	0.00316	-24.500
		C_{mqo}	-62.000	-60.000	0.01548	-60.000
		C_{mq2}	-175.000	-162.996	0.48505	-163.000
6	2-No Noise	ψ_o	0.5235	0.5235	0.00001	0.5235
		$\dot{\psi}_o$	0.0	0.0000	0.00014	0.0
		θ_o	0.5235	0.5235	0.00001	0.5235
		$\dot{\theta}_o$	0.0	.0000	0.00011	0.0
		$C_{m\bar{\alpha}}$	-1.800	-2.000	0.00001	-2.000
		C_{mqo}	-62.000	-59.998	0.00128	-60.000
7	3-No Noise	θ_o	0.5200	0.5235	0.00008	0.5235
		$\dot{\theta}_o$	0.0	0.0000	0.00112	0.0
		X_o	0.0	0.0002	0.00006	0.0
		$\dot{X}_o$	550.000	499.9993	0.00014	500.000
		Z_o	1050.000	999.9998	0.00004	1000.000
		$\dot{Z}_o$	0.0	0.0002	0.00006	0.0
		$C_{m\bar{\alpha}}$	-1.800	-2.000	0.00007	-2.000
		C_{mqo}	-64.000	-59.999	0.01661	-60.000
		C_{xo}	0.150	0.250	0.00001	0.250
8	3-No Noise	θ_o	0.1700	0.1745	0.00000	0.1745
		$\dot{\theta}_o$	0.0	0.0000	0.00006	0.0

RUN	DEGREES OF FREEDOM	PARAMETER EXTRACTED	INITIAL GUESS	EXTRACTED VALUE	ESTIMATED STANDARD DEVIATION	CORRECT VALUE
		X_o	0.0	0.0004	0.00025	0.0
		$\dot{X}_o$	500.000	701.1914	0.00073	701.19
		Z_o	1100.00	999.9993	0.00020	1000.000
		$\dot{Z}_o$	0.0	-.0010	0.00014	0.0
		$C_{m\bar{\alpha}}$	-11.160	-12.399	0.00002	-12.400
		C_{mqo}	-143.000	-131.142	0.00908	-130.00
		C_{xo}	0.200	0.174	0.00051	0.172
		$C_{z\bar{\alpha}}$	7.000	6.553	0.04904	6.400
9	3-No Noise	ψ_o	0.4000	0.3491	0.00000	0.3491
		$\dot{\psi}_o$	0.0	0.0000	0.00001	0.00
		Θ_o	0.4000	0.3491	0.00001	0.3491
		$\dot{\Theta}_o$	0.00	0.0000	0.00001	0.00
		ϕ_o	0.00	0.0002	0.00002	0.00
		$\dot{\phi}_o$	0.00	0.0011	0.00009	0.00
		$C_{\ell\bar{\alpha}}$	0.200	0.209	0.00001	0.209
		$C_{\ell p}$	-1.200	1.292	0.00057	-1.325
		$C_{m\bar{\alpha}}$	-3.000	-3.208	0.00004	-3.208
		C_{mq_o}	-30.000	-28.995	0.00515	-29.000
		$C_{np\bar{\alpha}}$	-7.000	-7.283	0.00219	-7.291
10	2-No Noise	Θ_o	1.500	1.5700	0.00000	1.5700
		$\dot{\Theta}_o$	0.00	0.000	0.0000	0.00
		ϕ_o	0.00	0.00	0.00000	0.00
		$\dot{\phi}_o$	0.900	1.000	0.00000	1.000
		$C_{\ell p}$	-1.200	-1.000	0.00005	-1.000

RUN	DEGREES OF FREEDOM	PARAMETER EXTRACTED	INITIAL GUESS	EXTRACTED VALUE	ESTIMATED STANDARD DEVIATION	CORRECT VALUE
		$C_{m\bar{\alpha}}$	-1.800	-2.000	0.00000	-2.000
		C_{mq_o}	-55.00	-60.000	0.00067	-60.000
11	6 - No Noise	ψ_o	0.1700	0.1743	0.00004	0.1745
		$\dot{\psi}_o$	0.000	-.0001	0.00060	0.00
		Θ_o	0.1700	0.1746	0.00004	0.1745
		$\dot{\Theta}_o$	0.00	-0.0011	0.00057	0.00
		ϕ_o	0.00	-.0001	0.00005	0.00
		$\dot{\phi}_o$	0.900	1.0006	0.00018	1.000
		X_o	0.00	0.0004	0.00020	0.00
		$\dot{X}_o$	550.00	499.998	0.00070	500.00
		Y_o	0.00	0.0000	0.00015	0.00
		$\dot{Y}_o$	0.00	0.0043	0.00028	0.00
		Z_o	1100.00	999.999	0.00018	1000.00
		$\dot{Z}_o$	0.00	0.0066	0.00019	0.00
		$C_{\ell p}$	-1.250	-.998	0.00038	-1.000
		$C_{m\bar{\alpha}}$	-1.750	-1.990	0.00009	-2.000
		C_{mq_o}	-55.000	-59.273	0.02063	-60.000
		C_{α_o}	-.200	0.249	0.00003	-.250
		$C_{y\bar{\alpha}}$	-1.000	-1.974	0.00147	-2.000
		$C_{z\bar{\alpha}}$	-1.000	-2.000	0.00031	-2.000

APPENDIX IV (Continued)

EXECUTION TIME OF TEST RUNS USING
AN IBM 370/165 DIGITAL COMPUTER

Run	Number of Iterations	Execution Time (Seconds)	RMSE
1	4	18.68	0.5484 x 10^{-2}
2	4	17.84	0.1216 x 10^{-2}
3	4	30.89	0.7027 x 10^{-2}
4	5	32.46	0.1507 x 10^{-3}
5	5	31.56	0.1504 x 10^{-3}
6	11	57.13	0.4616 x 10^{-3}
7	6	51.18	0.4259 x 10^{-3}
8_a	8		0.1529 x 10^{-1}
b	6		0.2061 [illegible]
c	4	105.99	0.1578 x [illegible]
9_a	6		0.1234 [illegible]
b	9		0.1452 [illegible]
c	2	106.91	0.8976 x [illegible]
10_a	3		0.2917 [illegible]
b	5	43.76	0.7741 [illegible]
11_a	6		0.5955 [illegible]
b	7		0.2702 x [illegible]
c	12	215.98	0.1291 [illegible]

REFERENCES

1. Chapman, G. T. and Kirk, D. B., "A Method for Extracting Aerodynamic Coefficients from Free Flight Data," AIAA Journal, Vol. 8, No. 4, pp. 753-758, April, 1970.

2. Bullock, T. E., Clarkson, M. H. and Daniel, D. C., "Preliminary Report on Extracting Aerodynamic Coefficients from Dynamic Data," Technical Report AFATL-TR-72-52, Air Force Armament Laboratory, March, 1972.

3. Holmes, J. E. and Woehr, F. A., "Wind-Tunnel Free-Flight Testing of Configurations with High-Fineness Ratio Bodies," AIAA 6th Aerodynamic Testing Conference, Paper No. 71-278, Albuquerque, March, 1971.

4. Meissenger, H. F., "The Use of Parameter Influence Coefficients in Computer Analysis of Dynamic Systems," Proceedings of the Western Joint Computer Conference, San Francisco, May, 1960.

5. Bullock, T. E., "Numerical Solution of a System of N First Order Differential Equations," Department of Electrical Engineering, University of Florida, 1968.

6. System 360 Scientific Subroutine Package, IBM Manual GH20-0205, August 1970, p. 118.

7. Etkin, Bernard, Dynamics of Flight, John Wiley & Sons, Inc., New York, 1959.

8. Goldstein, Herbert, Classical Mechanics, Addison-Wesley Publishing Co., Inc., Reading, Massachusetts, 1965.

(The reverse of this page is blank)

INITIAL DISTRIBUTION

ASD (ENYS)	1
AUL (AUL-LSE-70-239)	1
COMMANDER NWC	1
DDC	2
TRADOC/ADTC/DO	1
DL	1
DLB	1
DLOSL	2
USAFA/TECH LIBRARY	1
AFFDL	1
USN WEAPONS LAB	1
ARMY MSL COM/AMSMI-RDK	1
NASA AMES RSCH CENTER	1
RSCH LAB/AERO RSCH GP	1
USNOL	1
AEDC/ARO, INC.	1
AEDC/ARO, INC	1
AEDC/ARO, INC	1
ARMY MATERIEL ANAL AGCY	1
UNIV OF FLORIDA	5
DLD	1
DLDL	2
DLMA	15
HQ USAF/SAMI	1

(The reverse of this page is blank)

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) University of Florida Gainesville, Florida	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP ------

3. REPORT TITLE

A Digital Computer Program For Extracting Aerodynamic Coefficients From Six-Degree-Of-Freedom Dynamic Data

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Final Report - March 1972 to March 1973

5. AUTHOR(S) (First name, middle initial, last name)

F. W. Steinbauer
M. H. Clarkson
T. E. Bullock

6. REPORT DATE November 1973	7a. TOTAL NO. OF PAGES 80	7b. NO. OF REFS 7

8a. CONTRACT OR GRANT NO. F08635-73-C-0009	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO. 2069	
c. Task No. 01	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d. Work Unit No. 001	AFATL-TR-73-221

10. DISTRIBUTION STATEMENT

Distribution limited to U. S. Government agencies only; this report documents test and evaluation; distribution limitation applied November 1973. . Other requests for this document must be referred to the Air Force Armament Laboratory (DLMA), Eglin Air Force Base, Florida 32542.

11. SUPPLEMENTARY NOTES Available in DDC	12. SPONSORING MILITARY ACTIVITY Air Force Armament Laboratory Air Force Systems Command Eglin Air Force Base, Florida 32542

13. ABSTRACT

The development of a digital computer program to extract aerodynamic coefficients from dynamic data from six-degree-of-freedom systems is presented. The derivation of a system mathematical model is discussed in detail. Results, and associated problems, of extracting coefficients from one, two, three and six-degree-of-freedom systems data are also presented.

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Aerodynamic Coefficients						
Dynamic Data						
Six-Degree-of-Freedom Systems						